

CONTENTS

	Page
SUMMARY	1 1/A5
INTRODUCTION	1 1/A5
OVERVIEW	2 1/A6
Method of Analysis	2 1/A6
Blade geometric model	2 1/A6
Numerical model	3 1/A7
General Program Description	3 1/A7
DETAILED PROGRAM PROCEDURE	5 1/A9
Block Data NGASDAT	6 1/A10
Main Program NTTACT	6 1/A10
Subroutine FLOWS	7 1/A11
Subroutine FLSPLT	7 1/A11
Subroutine GASTBL	8 1/A12
Subroutine GAUSS	8 1/A12
Subroutine GETIN	8 1/A12
Function HCFRCD	9 1/A13
Subroutine HCOOL	9 1/A13
Subroutine HCPINS	9 1/A13
Subroutine INPRT	9 1/A13
Subroutine PARRAY	9 1/A13
Subroutine PLNUM	10 1/A14
Subroutine PLOTMF	10 1/A14
Subroutine PREP	10 1/A14
Subroutine TARRAY	10 1/A14
Subroutine TCOEF	11 1/B1
Subroutine THRCOM	11 1/B1
Subroutine WROUT	11 1/B1
DICTIONARY	11 1/B1
PROGRAM LISTING	53 1/E1
REFERENCES	150 2/E1

Item 830-H-15 NAS 1.60:1391

MAR 7 1979

NASA Technical Paper 1391

**COMPLETED
ORIGINAL**

**TACT1, A Computer Program for
the Transient Thermal Analysis
of a Cooled Turbine Blade or Vane
Equipped With a Coolant Insert**

II - Programmers Manual

Raymond E. Gaugler

JANUARY 1979

NASA

165

NASA Technical Paper 1391

**TACT1, A Computer Program for
the Transient Thermal Analysis
of a Cooled Turbine Blade or Vane
Equipped With a Coolant Insert**

II - Programmers Manual

Raymond E. Gaugler
Lewis Research Center
Cleveland, Ohio



National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1979

Blank Page

SUMMARY

A FORTRAN IV computer program to calculate transient and steady-state temperatures, pressures, and flows in a cooled turbine blade or vane with an impingement insert has been developed and is described in this report. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying one of three modes of heat transfer at each station: (1) impingement, including the effect of crossflow; (2) forced-convection channel flow; or (3) forced convection over pin fins. Additionally, a limited capability to handle film cooling is available in the program. It is assumed that spent impingement air flows in a chordwise direction and is discharged through a split or drilled trailing edge and through film-cooling holes. The program does not allow for radial flow of the spent impingement air. The use of film cooling is restricted by a numerical model requirement for a continuous coolant-channel flow.

Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The user can divide the blade by chordwise cuts into as many as 15 slices and can divide each slice into as many as 79 stations around the blade. Each station in turn consists of four calculational nodes through the wall and one in the coolant channel. The blade wall can be two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.

INTRODUCTION

As core turbine-engine operating conditions become more severe, it becomes more difficult to effectively cool blades and vanes. Advanced transient thermal calculational techniques are needed to design reliable turbine blades. However, there appears to be no computer program generally available that uses these advanced techniques in combining the required heat-transfer and coolant-flow-distribution calculations. Thus, it was decided to create a computer program that would perform both transient and steady-state heat-transfer and coolant-flow analyses for a cooled blade, given the outside hot-gas boundary conditions, the coolant inlet pressure or flow rate, the geometry of the blade shell, and the cooling configuration.

The resulting program, TACT1, can handle a turbine blade or vane that is equipped with a central coolant-plenum insert from which coolant air flows through holes to impinge on the inner surface of the blade shell or directly into the trailing-edge region. It

is assumed that the spent impingement air then flows chordwise and is dumped through a split or drilled trailing edge and through film-cooling holes. The blade is modeled by dividing it by chordwise planes into as many as 15 slices, with each slice having as many as 79 calculational stations around the blade. Temperatures at each station are calculated for four points through the wall and one in the coolant channel. Included in this model is the capability to analyze a blade with a ceramic thermal-barrier coating. The ability of the program to model film cooling is limited by the numerical flow analysis requirement for a continuous coolant-channel flow.

The TACT1 program is used at the NASA Lewis Research Center on an IBM TSS/360-67 computer. The source program consists of approximately 6000 lines of code and the program requires about 60 000 words of storage. Typical running times for the program are 1.4 seconds of central processor unit (CPU) time per calculational station for a steady-state run and 0.4 second of CPU time per station per time step for a transient run.

The TACT1 program is reported in two parts. This report, part II, is a programmers manual and includes a complete program listing and a detailed description of the procedure. Part I (ref. 1) is a users manual and contains all the information necessary to run the program: a detailed description of the input, the method of solution, and the output as well as a sample problem.

OVERVIEW

Method of Analysis

The details of the analytical method are presented in part I (ref. 1). The blade model used in the analysis is described briefly in this section.

Blade geometric model. - The key to creating a usable computer program is to have as simple a geometric model as possible for the system being analyzed. In this program, the emphasis is on a blade or vane with a central coolant plenum and chordwise flow of the coolant after impingement. Therefore, it was decided that the primary calculational direction would also be chordwise. The blade is divided into layers that are bounded by chordwise cuts through the blade, as shown in figure 1. Each slice is treated separately in the program, with radial heat conduction in the wall the only communication between layers.

Figure 2 gives the details of the geometric model for a single blade slice and shows the breakdown of the blade or vane into calculational stations and nodes. Each calculational station consists of five nodes: one at the wall outer surface, one at the interface between the coating and blade metal, one at a point midway through the wall metal, one at the wall inner surface, and one in the middle of the coolant channel.

For input to the program, the following basic elements of the geometry are needed for each station: (1) the thicknesses of the wall coating and wall metal and the coolant-channel width, (2) the distance of each node from the adjacent lower-numbered node, and (3) the radial span for this slice. In addition, depending on the mode of heat transfer specified, the user must supply impingement-hole diameter and spacing or pin-fin diameter and spacing. Thermal properties of the blade materials must also be specified. The input is described in detail in reference 1.

Numerical model. - The numerical solution for the temperatures throughout the blade involves writing a transient energy-balance equation for each node and forming a set of equations to be solved for the temperature distribution. Similarly, the coolant pressure distribution is determined by writing the transient momentum equation for flow between adjacent fluid nodes and solving the resulting set of equations for static pressures.

The nodal energy balances are linearized, one-dimensional heat-conduction equations at the wall outer-surface node, at the coating-metal interface, and at the wall inner-surface node. At the midmetal node, a linearized, three-dimensional, heat-conduction equation is used. In the coolant channel, energy and momentum equations for one-dimensional compressible flow including friction and heat transfer are written for the elemental channel length between two coolant nodes. The equations used are presented in reference 1.

General Program Description

The TACT1 program is capable of performing a transient analysis as well as a steady-state analysis. In the case of a transient, the program first performs steady-state calculations to determine the initial conditions for the transient.

Figure 3 shows a schematic of the TACT1 calculational scheme. There are three basic, nested calculational loops that must converge for a steady-state solution to be reached. These loops are labeled A, B, and C in figure 3. The program begins a steady-state analysis with the coolant-supply pressure and total coolant flow fixed. The impingement flow is initially assumed to split uniformly at the leading-edge stagnation station, station 1. All coolant flows for the slice under consideration are calculated first, based on the current pressure distribution. The temperatures at each node are then calculated by solving simultaneously the energy equations presented in reference 1. The pressures at each coolant node are calculated by solving simultaneously the momentum equations presented in reference 1. This cycle, loop A in figure 3, is repeated until the pressure distribution no longer changes. The flow split between suction- and pressure-side coolant channels is then checked by comparing the pressures at the ends of the two channels. If they do not match, the impingement flow split at the leading edge

is adjusted and the inner loop calculations are repeated. This adjustment comprises loop B in figure 3. Once the proper flow split is achieved, the program moves up the blade to the next slice and repeats this sequence. After all N slices have converged, the total coolant mass flow used is compared with the inlet coolant mass flow. If there is an imbalance, either the inlet flow or the supply pressure is adjusted, depending on which was specified in the input; and the calculations start over. This is loop C in figure 3. Once the overall coolant mass flow balance is satisfied, the steady-state solution is complete and the transient calculations begin. During a transient calculation, loop B is bypassed because the coolant flow-split is primarily a function of blade geometry. Loop C is also bypassed because the inlet coolant mass flow rate at a given time is estimated based on the coolant mass flow used at the previous time step and the change in supply pressure.

The TACT1 subprograms and the calling relations are shown in figure 4. Block data subprogram NGASDAT contains air properties, for use in TACT1, tabulated as functions of temperature at a pressure of 20 atmospheres from reference 2. This subprogram must be loaded before execution of the program. The main program, NTTACT, calls other subroutines in their proper order.

The first call from NTTACT is to GETIN, a subroutine that controls the reading, storing, and printing of input data. Subroutine GETIN calls INPUT to print the input data if the user specifies INEDIT > 0. Subroutine INPRT has a call to PREP to put the input data in its proper form for use. All data are input by using a NAMELIST format.

After the input data have been read, the number of time steps, NTYM, to be used in the transient is determined in NTTACT. If only a steady-state solution is to be calculated, NTYM = 1. Time-dependent boundary conditions are then evaluated, with the initial entries assumed to be steady-state values. Then NTTACT loops through the blade, calling on subroutines PLNUM, PREP, and TCOEF for each slice. The first time through is a steady-state calculation.

Subroutine PLNUM calculates the pressure distribution in the impingement plenum for the current slice, given the inlet pressure and coolant flow-rate. PLNUM calls GASTBL for gas properties.

Subroutine PREP extracts the input data for the current slice from the input arrays.

Subroutine TCOEF controls loop A in figure 3, the iterative calculations of temperature and pressure for the nodes of the current slice. Each iteration in TCOEF requires calls to subroutines FLOWS, HCOOL, THRCOM, TARRAY, PARRAY, and GAUSS.

Subroutine FLOWS computes the impingement jet flow rates, coolant-channel mass flow rates, and channel Mach numbers for each station around the blade, given the plenum pressure and temperature and the current pressure distribution in the coolant channel. FLOWS calls GASTBL for gas properties.

Subroutine HCOOL is called to calculate coolant heat-transfer coefficients for all the stations of this slice, based on the latest values of mass flow rate. HCOOL calls function HCFRCD to calculate forced-convection heat-transfer coefficients and GASTBL for gas properties.

Subroutine THRCON determines the wall thermal conductivity from the input table of conductivity as a function of temperature.

Subroutine TARRAY sets up the array of coefficients for the conduction and convection equations for each node. Calls are made to HCPINS for pin-fin heat-transfer coefficients, to HCFRCD for forced-convection heat-transfer coefficients, and to GASTBL for gas properties. TCOEF calls subroutine GAUSS to solve the set of equations for the temperature at each node.

Subroutine PARRAY sets up the array of coefficients for the momentum equations in the coolant channels and TCOEF calls subroutine GAUSS to solve the set of equations for the pressure at each coolant node.

After a new set of temperatures and pressures has been determined, convergence is checked by using the coolant-channel pressure at the blade leading edge. If this pressure stays within a tolerance band for four successive iterations, convergence is accepted. Once convergence is achieved, TCOEF calls subroutine FLSPLT to check the coolant flow-split between the pressure and suction sides. This is loop B in figure 3. Initially, the impingement jet flow at the forward stagnation station is assumed to split evenly between the suction- and pressure-side channels. If the coolant-channel pressures at the end of the impingement insert do not match, the flow split at the forward stagnation station is adjusted to increase the flow to the channel with the higher pressure at the end of the insert, and iteration loop A is repeated. Once a satisfactory flow split has been achieved, TCOEF calls subroutine WROUT to print the output for this slice and calls subroutine PLOTMF if there is to be graphical output. After NTTACT has calculated all blade slices, the total coolant mass flow is compared with the impingement-plenum inlet mass flow rate used to start the calculations. If the two flow rates are not close enough, the inlet mass flow or supply pressure is adjusted and the calculations are repeated. This is loop C in figure 3.

When the initial steady-state solution has been completed, the transient calculations are started. The transient is continued until the time reaches the specified maximum.

Subroutine PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center to plot temperature and pressure distributions for the blade.

DETAILED PROGRAM PROCEDURE

Table I lists the names of each of the subprograms in TACT1, the corresponding

TSS/360 source module names, the COMMON blocks used in each, the names of the subroutines called by each, and the names of subroutines calling each. Table II is a cross-reference listing of named COMMON blocks and the subprograms using them. This section gives a detailed description of each subprogram used in TACT1. All variable names used are defined in the section DICTIONARY. The BLOCK DATA subprogram and the MAIN PROGRAM are discussed first and then each subprogram is described, in alphabetical order.

Block Data NGASDAT

A BLOCK DATA subprogram, NGASDAT, is used to provide a table of gas properties to the program. The properties are put in the array GS through a DATA statement with $5 \times NG$ entries, where NG is the number of table entries for each property. The first NG values are temperatures, the second are thermal conductivities, the third are specific heats, the fourth are Prandtl numbers, and the final NG values are viscosities. The property values included are taken from reference 2 at a pressure of 20 atmospheres.

Main Program NTTACT

The MAIN PROGRAM for TACT1, NTTACT, has overall control of the program. Figure 5 is a flow chart for NTTACT. During initialization, a call is made to a system subroutine, TIME, to get a unique label to be used to identify the plotted output for a given run. After the call to GETIN, where all the input data are read, NTTACT initiates the solution procedure by searching the transient boundary condition tables and using linear interpolation to extract the values for the current time. The next step is to begin the loop, labeled C in figure 3. The solution progresses from hub to tip. For each slice, NTTACT calls PLNUM to calculate coolant-supply conditions; PREP to extract the input data from the input tables; and TCOEF to calculate flows, temperatures, and pressures. After the return from TCOEF, NTTACT updates the total amount of coolant used, WUSED, by adding the amount used in the current slice, WIM. The amount of coolant-plenum flow available for the remaining slices, WPLEN, is updated by subtracting WIM. After all slices have been done, the overall amount of coolant used is printed and then checked against the assumed coolant flow-rate. If the absolute value of the difference, EXCESW, is more than 1 percent of the assumed flow, the assumed flow or the supply pressure is adjusted and the calculations are repeated. For transient runs, after the initial steady-state coolant-flow balance, there are no more iterations on coolant flow. Instead the flow for a given time step is based on the actual flow used in

the preceding time step and on the ratio of supply pressure for the two steps. Finally, once all loops have been completed, NTTACT calls PLOTMF to get a final summary plot of blade temperatures.

Subroutine FLOWS

Subroutine FLOWS is a routine to calculate the flow rates through all impingement and film-cooling holes, the friction factor in the coolant channels, and film-cooling effectiveness. FLOWS makes use of the current impingement-plenum mean pressure and temperature and coolant-channel pressure and temperature distributions. The impingement jet flow-rate, WJ, is calculated for each station in the forward region and checked against the choked flow-rate, WCR. If WJ is greater than WCR, then WJ is set equal to WCR. If there is any film cooling on the blade, the film-cooling flow rates in the forward region, WFC, are also calculated. Then, the coolant-channel flow rates, WCROS, are computed by considering a mass balance between stations, as illustrated in figure 6. Once the forward-region coolant flows have been determined, the Reynolds numbers - RE for the coolant channel, and REFC for the film-cooling flow - and the square of the coolant Mach number, AM2, are computed for each forward station.

The next step is to calculate the amount of coolant, WDUMP, dumped directly into the trailing-edge region from the coolant plenum. Then the total amount of coolant used for this slice, WIM, is determined by summing the impingement jet flows and WDUMP. Following this, the flows in the trailing-edge region are computed, with the coolant flow being reduced by the amount of any film-cooling flow. Then, trailing-edge-region values of RE, REFC, and AM2 are calculated.

After all the coolant flow-rates are determined, the friction factor, FF, is calculated at each station. Finally, if there is any film cooling used, the film effectiveness is calculated by using the method of reference 3.

Subroutine FLSPLT

Subroutine FLSPLT is used to determine the location of the stagnation impingement jet, station JS, and the fraction of that jet's flow that splits to each side of the blade, DELTAN. Figure 7 is a detailed flow chart for subroutine FLSPLT. The primary variable carried into FLSPLT is the pressure-match parameter, EPSN, which is defined as

$$\text{EPSN} = \frac{(P(2, \text{ISLICE}, \text{NFWD}-1) - P(2, \text{ISLICE}, \text{NFWD}))}{P(2, \text{ISLICE}, \text{NFWD}-1)} \quad (1)$$

where the pressures are as illustrated in figure 8.

The magnitude and sign of EPSN are used to determine the adjustment of the stagnation impingement-jet row location and the fraction of that jet that splits to the suction-side channel. Initially, the stagnation jet row is located at station 1 and the split is DELTAN = 0.50. If EPSN is positive, DELTAN is set to 0.75 to increase the flow down the suction-side channel; if EPSN is negative, DELTAN is set to 0.25 to increase the flow down the pressure-side channel. For subsequent entries into FLSPLT, the value of DELTAN is adjusted by passing a straight line through the last two points on a plot of EPSN versus DELTAN and picking the value of DELTAN where this line crosses the axis at EPSN = 0. If this intercept falls outside the DELTAN range of 0 to 1, the stagnation station, JS, must be moved to an adjacent station and DELTAN set to 0.50. Once a sign change is observed in EPSN, a fine-tuning process is triggered in FLSPLT. In this case, the values of DELTAN and EPSN for the iteration preceding the sign change are saved and used as one of the points of the straight-line interpolation scheme for all subsequent iterations.

Subroutine GASTBL

Subroutine GASTBL is used to interpolate in the array GS for gas properties, given the absolute temperature. Linear interpolation is used.

Subroutine GAUSS

Subroutine GAUSS is a routine that uses Gaussian elimination to solve a set of simultaneous equations. The array of coefficients, TCOF, is in the form of a compressed, augmented band matrix. That is, only the matrix elements within the band and the constants from the right side are stored in TCOF. The matrix band width, K, is determined by the node-numbering system used. In TACT1, the temperature calculations require a band width of 23 elements, and the pressure calculations require 19.

Subroutine GETIN

Subroutine GETIN is a routine used to initialize input-data default values and to read and store input data. Input is in NAMELIST form as described in reference 1. The entire data set is read, and the input variables for each slice are stored in two arrays: INDCHN for integer data, and CHANL for real-number data. If the input is provided in SI units, subroutine GETIN converts it to U.S. customary units for internal use. If the user specifies INEDIT > 0, GETIN calls subroutine INPRT to print out the input data.

Function HCFRCD

Function subprogram HCFRCD is a routine to calculate a turbulent, forced-convection heat-transfer coefficient for channel flow as described in reference 1.

Subroutine HCOOL

Subroutine HCOOL is a routine containing the correlations for impingement heat transfer. The first part of HCOOL deals with leading-edge-region impingement cooling. In this part, the inner surface length from the stagnation impingement jet to the end of the leading-edge impingement region is determined and then used in a correlation to compute the average heat-transfer coefficient in this region. Beyond this region, for stations starting at ICOR, calculations are done by using an impingement-with-crossflow correlation.

Subroutine HCPINS

Subroutine HCPINS is a routine to calculate coolant-side heat-transfer coefficients in regions of the blade equipped with pin fins. In addition, the effective heat-transfer area, which accounts for the pin surface area and the pin-fin effectiveness, is calculated.

Subroutine INPRT

Subroutine INPRT is a routine to print a listing of the input data. Also, INPRT sets up the initial temperature distribution in the blade. Subroutine PREP is called for each slice to extract input data from the arrays INDCHN and CHANL.

Subroutine PARRAY

Subroutine PARRAY is a routine to set up the matrix to be solved for coolant-channel pressure distribution. The equations used are detailed in reference 1.

The array of coefficients generated in PARRAY, TCOF, is in the form of a compressed, augmented band matrix. Coefficients that would be on the main diagonal of the full matrix are stored in column 10 of the TCOF array. The terms from the right side of the equations are stored in column 20 of TCOF.

Subroutine PLNUM

Subroutine PLNUM is a routine to calculate the pressure and temperature distributions in the central coolant plenum. The mean plenum static pressure and temperature for each slice are used as the supply conditions for the impingement jets. The total temperature and pressure at the outlet of one plenum slice are used as input for the next slice.

There are five arguments used in the call statement for PLNUM: WXX is the mass flow rate into this plenum slice; PXX and TXX are the calculated, average static pressure and temperature; and PTEXIT and TTEXIT are total temperature and pressure, respectively. Going into the subroutine, PTEXIT and TTEXIT are the values at the entrance to this slice. On return, they are the values at the exit of this slice.

Subroutine PLOTMF

Subroutine PLOTMF is a routine that plots TACT1 output. PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center. For an installation without this specific package, this subroutine would have to be revised or bypassed.

PLOTMF plots temperature and pressure versus surface distance from station 1 for each slice of the blade for a steady-state case. For transients, a set of two summary plots is made for each time step: the plots contain temperatures for all slices on one graph.

Subroutine PREP

Subroutine PREP is a routine to extract input data from storage and put it in the form used in the calculations. In PREP, the hot-gas-side boundary condition tables are searched and linear interpolation is done to extract the boundary condition values at each calculation station at the given time.

Subroutine TARRAY

Subroutine TARRAY is a routine to set up the matrix to be solved for the temperatures in each slice. The equations used are detailed in reference 1.

The array of coefficients generated in TARRAY, TCOF, is in the form of a compressed, augmented band matrix. The 12th column of TCOF contains the elements that would be on the main diagonal of a full matrix. The terms from the right side of the equations are stored in column 24.

Subroutine TCOEF

Subroutine TCOEF is a routine that controls the calculations for flow rates, temperatures, and pressures. The first time TCOEF is entered for each slice an initial estimate of the coolant-channel pressure distribution is set up. TCOEF controls the iterations in loops A and B in figure 3. Loop A consists of calls to subroutines FLOWS, TARRAY, and PARRAY. The variable IVERGE is used to count the number of iterations in loop A. Convergence is checked by comparing the four most-recent values of coolant-channel pressure at the flow-split point. When the ratio of the maximum difference among these four to the difference between coolant-supply pressure and trailing-edge exit pressure is less than PCNVRG, loop A is complete. Then the flow split at the stagnation impingement jet, JS, is checked by subroutine FLSPLT and adjusted if necessary. Loop B involves repeating loop A for a new flow split. The variable IDELT is used to count the number of flow-split iterations in loop B. Once flow-split convergence is achieved, WROUT is called to print the output for the current slice.

Subroutine THRCON

Subroutine THRCON is a routine that takes the wall temperatures and searches for the thermal conductivity values in the input tables.

Subroutine WROUT

Subroutine WROUT is a routine to control the printing of the output from TACT1. Output units are the same as the input data units.

DICTIONARY

All the important variable names used in the TACT1 code are defined in this section. The only names not defined are locally used indices. All dimensioned variables include the dimensions. The dictionary also indicates the COMMON block or subroutine in which each variable is used.

Variable	Common	Subroutine	Definition
A(400)	TCO		cross-sectional area normal to chord-wise direction, in ² , accessed by node number
AA		GETIN	outer-surface length between stations, in., used for calculating interpolated values of TDLX(2), TDLX(3), and TDLX(5)
AA		PLNUM	coolant-plenum cross-sectional area, in ² , used in plenum pressure-drop calculations
AB		PLNUM	maximum Mach number in coolant plenum
AC(5)		GASTBL	array of interpolated values of gas properties
ACH		PLNUM	coolant-plenum choked-flow indicator
ADUMP	TCO		area of slot or jets dumping coolant directly into trailing-edge region, in ²
ADUMPC		INPRT	same as ADUMP, but converted to cm ² for input listing when input is in SI units
AHG		PREP	intermediate value of hot-gas-side heat-transfer coefficient, Btu/hr . ft ² . °F, used for interpolating in input table during a transient
AHTRN1		TARRAY	inner-surface area for heat-transfer purposes, in ²
AHTTR		HCPINS	total surface area in pin-fin channel, in ²
AINTRV		PLOTMF	floating-point form of number of temperature intervals in summary plots
AJ		PLNUM	floating-point form of indicator J - 1
AJET(80)	TCO		total area of impingement jets at each station, in ²

Variable	Common	Subroutine	Definition
AJET(80)		FLSPLT	total area of impingement jets at each station, in ² , carried into subroutine as argument
AKC(15,80)	TCO		wall outer-coating thermal conductivity, Btu/hr . ft . °F
AKCTBL(20)	BOUND		input table of wall outer-coating thermal conductivity, Btu/hr . ft . °F, versus temperature, °F
AKW(15,80)	TCO		wall metal thermal conductivity, Btu/hr . ft . °F
AKWTBL(20)	BOUND		input table of wall metal thermal conductivity, Btu/hr . ft . °F, versus temperature, °F
ALABL(7)		PLOTMF	array containing time and date label for identification of output plots
ALPH(12)		NTTACT	alphameric array used to uniquely identify output of each job
ALPHA	FRIC		constant used in friction factor calculations
ALPH2(4)		PLOTMF TCOEF NTTACT	time and date information, generated in NTTACT and passed to plotting subroutine as argument
AM	HCOOL		exponent on Reynolds number in Kercher-Tabakoff impingement correlation
AMC(20)	PLNUM		Mach number distribution in coolant plenum for a given slice
AMCHOK	FLOWs		if any stations have a Mach number greater than 1.0, the value is saved in this variable and returned as an argument, to be printed by TCOEF
AMIN	FLOWs HCPINS		area of coolant-flow channel at a given station, reduced by pin-fin blockage

Variable	Common	Subroutine	Definition
AM2(80)	TCO		array containing square of coolant-channel Mach number at each station, for a given slice
APG		PREP	intermediate value of hot-gas-side pressure, lbf/in ² , used for interpolating in input table during transient
APLEN		GETIN	input value of coolant-plenum area for given slice, cm ² (in ²)
APLN(15)	RADL		internal array to store plenum area for each slice, in ²
AP1		GASTBL	interpolating factor in gas property table lookup
AP2		GASTBL	1.0 - AP1
AQG		PREP	intermediate value of hot-gas-side heat flux, Btu/hr · ft ² , used for interpolating in input table during transient
ASTG		TCOEF	inner-surface area under stagnation-point impingement jet, in ²
ATG		PREP	intermediate value of hot-gas-side temperature, °R, used for interpolating in input table during transient
ATMAXP		PLOTMF	adjusted maximum temperature, °F, used as high endpoint on output plots
ATMINP		PLOTMF	adjusted minimum temperature, °F, used as low endpoint on output plots
ATYME		PLOTMF	value of time in transient, sec, used on output plots for identification
AVRGA		PARRAY	area ratio used in momentum equation at entrance to trailing edge
AZ		PLNUM	dummy variable, used as either diameter-area ratio or flow adjustment

Variable	Common	Subroutine	Definition
A1		FLOWs	interpolating factor in friction factor calculation in transitional Reynolds number range
A1		TARRAY	upstream half of inside-wall heat-transfer area, in ² , associated with coolant-channel node
A2		FLOWs	interpolating factor in friction factor calculation in transitional Reynolds number range
A2		TARRAY	downstream half of inner surface heat-transfer area, in ² , associated with coolant-channel node
A3		TARRAY	same as A1, but on opposite wall, only used in trailing-edge region
A4		TARRAY	same as A2, but on opposite wall, only used in trailing-edge region
B		GETIN	ratio of length to thickness, used along with AA to calculate interpolated values of TDLX(2), TDLX(3), and TDLX(5)
B(20)		PLNUM	spanwise static-temperature distribution, °R, in coolant plenum for given slice
BC		GETIN	NAMELIST name
BCHGP(1000)	BOUND		input table of hot-gas, pressure-side heat-transfer coefficients, W/m ² · K (Btu/hr · ft ² · °F)
BCHGS(1000)	BOUND		input table of hot-gas, suction-side heat-transfer coefficients, W/m ² · K (Btu/hr · ft ² · °F)
BCPGP(1000)	BOUND		input table of hot-gas, pressure-side relative static pressure, kPa (lbf/in ²)
BCPGS(1000)	BOUND		input table of hot-gas, suction-side relative static pressure, kPa (lbf/in ²)

Variable	Common	Subroutine	Definition
BCQGP(1000)	BOUND		input table of hot-gas, pressure-side heat flux, W/m^2 ($\text{Btu/hr} \cdot \text{ft}^2$)
BCQGS(1000)	BOUND		input table of hot-gas, suction-side heat flux, W/m^2 ($\text{Btu/hr} \cdot \text{ft}^2$)
BCTGP(1000)	BOUND		input table of hot-gas, pressure-side temperature, K ($^{\circ}\text{F}$)
BCTGS(1000)	BOUND		input table of hot-gas, suction-side temperature, K ($^{\circ}\text{F}$)
BCTIME(50)	BOUND		input table of time at which transient input tables are specified, sec
BCXP(400)	BOUND		input table of outer-surface, pressure-side locations at which hot-gas conditions are input, cm (in.)
BCXS(400)	BOUND		input table of outer-surface, suction-side locations at which hot-gas conditions are input, cm (in.)
BES		HCOOL	equivalent slot width, in., used in leading-edge impingement correlation
BETA	FRIC		constant used in friction factor calculations
BETA1		PLNUM	square of pressure at inlet to coolant plenum for given slice, $(\text{lbf/in}^2)^2$
BETTA(20)		PLNUM	spanwise static-pressure distribution in coolant plenum, lbf/in^2
BT A	TCO		indicates type of hot-gas boundary condition
C		FLOWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	gas thermal conductivity, $\text{Btu/hr} \cdot \text{ft} \cdot {^{\circ}\text{F}}$

Variable	Common	Subroutine	Definition
CD	TCO		impingement-jet discharge coefficient
CDEN(2)	UNITS		conversion factor for density units
CD1(200)		NTTACT	dummy variable used to print selected intermediate temperature values
CEXCSW		NTTACT	amount of excess coolant flow, in SI units, kg/hr
CGASC(2)	UNITS		conversion factor for gas constant
CH(15)		PLNUM	coolant-channel choking indicator
CHANL(8000)	SPECL		array for storing input data
CHANLS		GETIN	NAMELIST name
CHFLX(2)	UNITS		conversion factor for heat-flux units
CHTC(2)	UNITS		conversion factor for heat-transfer-coefficient units
CIMP1	IMPCOR		user-supplied constants for general impingement correlation
CIMP2			
CIMP3			
CIMP4			
CIMP5			
CIMP6			
CIMP7			
CINCH(2)	UNITS		conversion factor for length units
CMSFL(2)	UNITS		conversion factor for mass flow rate units
CNUM(80)	TCO		number of impingement jets at each station for given slice
CONDCT		HCOOL	coolant-air thermal conductivity, Btu/hr · ft · °F
CONTRL		GETIN	NAMELIST name
CP	TCO		gas specific heat at constant pressure, Btu/lbm · °F

Variable	Common	Subroutine	Definition
CPC(80)	PRPS		coolant specific heat at constant pressure at each coolant node for given slice, Btu/lbm · °F evaluated at a mean temperature between bulk coolant temperature and wall temperature
CPIM		NTTACT	mean impingement-plenum pressure for given slice, in SI units, kPa
		PLOTMF	
CPM		FLOWS	hot-gas-stream specific heat at constant pressure, Btu/lbm · °F
CPO	PRPS		specific heat at constant pressure, Btu/lbm · °F, evaluated at impingement-jet supply temperature
CPRSR(2)	UNITS		conversion factor for pressure units
CRHOVG(2)	UNITS		conversion factor for density × velocity units
CRITR		FLSPLT	coolant flow-split convergence criterion
CSPHT(2)	UNITS		conversion factor for specific-heat units
CTCON(2)	UNITS		conversion factor for thermal conductivity units
CTMPF(2)	UNITS		conversion factor for temperature units
CT0G		NTTACT	mean impingement-plenum static temperature for given slice, in SI units, K
CURV		TARRAY	factor to account for wall curvature in heat-conduction equations
CVISC(2)	UNITS		conversion factor for viscosity units
CWPLEN		NTTACT	coolant-plenum flow rate at entrance to given slice, in SI units, kg/hr
CWUSED		NTTACT	total amount of coolant air used, in SI units, kg/hr
CX		PLNUM	function of isentropic exponent k , $-(k + 1)/[2(k - 1)]$

Variable	Common	Subroutine	Definition
C1		PLNUM	function of isentropic exponent k, $2k/(k - 1)$
C3		PLNUM	computed constant involving wheel speed and isentropic exponent
C3		FLOWS	ratio of specific heats at constant pres- sure, coolant to hot gas
C5		PLNUM	computed constant involving isentropic exponent and gas constant
C6		PLNUM	function of isentropic exponent k, $(k - 1)/2$
C7		PLNUM	computed constant involving isentropic exponent and gas constant
C8		PLNUM	computed constant involving isentropic exponent and gas constant
D		PLNUM	convergence parameter in coolant- plenum pressure calculations
DD		PLNUM	coolant-plenum hydraulic diameter, in
DEH		HCOOL	hydraulic diameter of equivalent slot, in., used in leading-edge impingement correlation
DELAST		FLSPLT	variable used to save flow-split fraction at which pressure-match parameter, EPSN, changes sign
DELTA	FRIC		constant used in friction factor calcula- tion
DELTAN(15)		FLOWS FLSPLT HCPINS TARRAY TCOEF WROUT	fraction of stagnation-point impingement- jet flow that splits to suction-side coolant-flow channel for each slice
DELTAO		FLSPLT	value of DELTAN from previous flow- split iteration

Variable	Common	Subroutine	Definition
DENOM		NTTACT	intermediate variable used in time interpolation of some boundary conditions
DH(80)	TCO		coolant-channel hydraulic diameter at each station, in.
DHF(80)	TCO		effective diameter of film-cooling hole at each station, in., defined as hydraulic diameter of one hole multiplied by square root of number of holes at station
DHJ(80)	TCO		actual hydraulic diameter of an impingement hole at each station, in.
DHYD		GETIN	input value of coolant-plenum hydraulic diameter for a slice, cm (in.)
DIFN		TCOEF	pressure difference parameter used in checking convergence
DIFO		TCOEF	maximum pressure difference parameter used in checking convergence
DIFTOL		PLNUM	tolerance on coolant-plenum pressure-drop calculations
DIMP1	IMPCOR		user-supplied constants for leading-edge impingement correlation
DIMP2			
DIMP3			
DIMP4			
DIMP5			
DIMP6			
DLTAOP		FLSPLT	best value of DELTAN in the event of an unstable flow split
DLTYME	TRNSNT		time step used in transient calculations, sec
DLX(400)	TCO		chordwise distance from each node to adjacent upstream node, in.
DP(80)	PRPS		diameter of pin fins at each station, in.

Variable	Common	Subroutine	Definition
DPLN(15)	RADL		coolant-plenum hydraulic diameter for each slice, in.
DR		PLNUM	radial length increment in coolant-plenum calculations, in.
DR2		PLNUM	radial increment squared, in ²
DUMR1(80)	PRPS		dummy variable, not currently used
DUMR2(80)	PRPS		summary variable, used to carry impingement-jet Reynolds number to output subroutine
DUMTER		PARRAY	intermediate variable in momentum equation involving coolant dumped directly into trailing edge
DUM1(10)		INPRT	dummy variables used to print input
		WROUT	listings and program output
DUM2(10)		INPRT	
		WROUT	
DUM3(10)		INPRT	
DUM4(10)			
DUM5(10)			
DUM6(10)			
DUM7(10)			
DUM8(10)			
DUM9(10)			
DUM10(10)			
DUM11(10)			
DUM12(10)			
DUM13(10)			
DUM14(10)			
DUM15(10)			
DUM16(10)			

Variable	Common	Subroutine	Definition
DUM17(10)		INPRT	dummy variables used to print input listings and program output
DUM18(10)			
DUM19(10)			
DUM20(10)			
DUM25(10)			
DUM52(10)			
DUM53(10)			
DUM55(10)			
DX		PLNUM	spanwise step size used in calculating coolant-plenum pressure and temperature distributions
DXTEMP		PLNUM	variable to temporarily hold DX
DX1		TARRAY	path length between midwall node and adjacent upstream midwall node, in.
DX10		TARRAY	path length between outer coating - wall junction node and adjacent downstream outer coating - wall junction node, in.
DX2		TARRAY	path length between midwall node and adjacent downstream midwall node, in.
DX3		TARRAY	path length between outer-surface node and adjacent upstream outer-surface node, in.
DX4		TARRAY	path length between outer-surface node and adjacent downstream outer-surface node, in.
DX5		TARRAY	path length between inner-surface node and adjacent upstream inner-surface node, in.
DX6		TARRAY	path length between inner-surface node and adjacent downstream inner-surface node, in.

Variable	Common	Subroutine	Definition
DX7		TARRAY	path length between coolant node and adjacent upstream coolant node, in.
DX9		TARRAY	path length between outer coating - wall junction node and adjacent upstream outer coating - wall junction node, in.
D1		PLNUM	computed constant used in coolant-plenum pressure equations to account for effect of pumping due to wheel rotation
D2		PLNUM	computed constant used in coolant-plenum temperature equations to account for effect of pumping due to wheel rotation
E		PLNUM	factor used in adjusting convergence rate in coolant-plenum calculations
EFAREA(80)		HCPINS TARRAY	effective area, in ² , for heat transfer at stations with pin fins, including pin-fin effectiveness for heat transfer
EFTVNS		HCPINS	pin-fin effectiveness
EMES(80)	FLMCOL		for film cooling, ratio of coolant mass flux to free-stream mass flux, multiplied by equivalent slot width
EML		HCPINS	term used in pin-fin effectiveness calculation
ENDEFF		TARRAY	term in heat-transfer equations to account for convection to rear edge of blade when heat-transfer coefficients are input
ENDFLX		TARRAY	term in heat-transfer equations to account for convection to rear edge of blade when heat flux is input
EPLAST		FLSPLT	variable used to save latest value of pressure-match parameter, EPSN

Variable	Common	Subroutine	Definition
EPS	FRIC		constant used in friction factor calculations
EPSMIN		FLSPLT	minimum value attained by pressure-match parameter, EPSN, for unstable flow split
EPSN		FLSPLT	pressure-match parameter, defined as difference between suction- and
	TCOEF		pressure-side coolant-channel static pressures at end of insert, divided by suction-side coolant-channel static pressure
EPSO		FLSPLT	old value of pressure-match parameter, EPSN
ETAPRM		FLOWS	film-cooling effectiveness based on ratio of enthalpy differences
EXCESW		NTTACT	amount of excess coolant flow, difference between inlet flow and that actually used, lbm/hr
FACTOR		TARRAY	special variable to adjust amount of energy carried in by an impingement jet for case of a calculation station adjacent to flow-split station
FF(80)	TCO		value of friction factor at each flow station
FILM		PARRAY	term to account for momentum carried off by film-cooling air
FILMW		TARRAY	total film-cooling flow from given coolant node
FLMEFF(80)	FLMCOL		film-cooling effectiveness based on ratio of temperature differences
FM		GAUSS	multiplying factor used in Gauss elimination scheme

Variable	Common	Subroutine	Definition
FUNP		PLNUM	statement function to calculate pressure difference in coolant plenum
FUNT		PLNUM	statement function to calculate temperature difference in coolant plenum
F1(20)		PLNUM	friction factor in coolant plenum
GAM	TCO		ratio of specific heats
GAMC(80)	PRPS		ratio of specific heats at each coolant-channel node
GAMO	PRPS		ratio of specific heats at coolant-supply conditions
GEO		GETIN	NAMELIST name
GG		HCOOL	mass flux ratio, coolant crossflow to impingement-jet flow
GI		HCOOL	momentum flux ratio, coolant crossflow to impingement-jet flow
GMASS		HCOOL	mass flux from row of leading-edge impingement holes
GS(200)	GAAS		table of gas properties
G1		PLNUM	computed constant in coolant-plenum calculations, involving flow rate, gas constant, and specific heat at constant pressure
G2		PLNUM	computed constant in coolant-plenum calculations, involving flow rate and gas constant
HBAR		WRROUT	average coolant-side heat-transfer coefficient for given slice, $\text{W}/\text{m}^2 \cdot \text{K}$ ($\text{Btu}/\text{hr} \cdot \text{ft}^2 \cdot {}^\circ\text{F}$)
HC(80)	TCO		coolant-side heat-transfer coefficients at each station for given slice, $\text{Btu}/\text{hr} \cdot \text{ft}^2 \cdot {}^\circ\text{F}$

Variable	Common	Subroutine	Definition
HCAL(4)		INPRT	alphameric array containing labels identifying type of coolant-side heat transfer
HG(80)	TCO		hot-gas-side heat-transfer coefficient at each station for given slice, Btu/hr · ft ² · °F
HSTGMX		TCOEF	maximum physically possible value of coolant heat-transfer coefficient under stagnation jet, Btu/hr · ft ² · °F
HUB1		TARRAY	term in conduction equation to account for specified hub temperature
HUB3		TARRAY	term in conduction equation to account for specified hub heat flux
HX		TARRAY	multiplying factor on coolant heat-transfer coefficient, initialized to 1.0 but may be changed dynamically
HYCOS		HCPINS	hyperbolic cosine term
HYSIN		HCPINS	hyperbolic sine term
IADJIN	SPECL		input variable that indicates which coolant-plenum supply property is to be held fixed
ICHK		PARRAY	indicates which side of blade a given station is on: 0 if suction side, 1 if pressure side
ICHNL		INPRT PREP	slice number, carried through as argument
ICHOKE		FLOWs PARRAY TCOEF	number of station that shows choked coolant flow
ICOMP		TARRAY	number of station adjacent to impingement flow-split station in pressure-side direction

Variable	Common	Subroutine	Definition
ICOMS		TARRAY	number of station adjacent to impinge- ment flow-split station in suction-side direction
ICONV		FLSPLT TCOEF	indicator for convergence of flow-split iterations
ICOR	TCO		station at which use of impingement- with-crossflow correlation is to begin
IDELT		FLSPLT TCOEF WRROUT	counter of number of flow-split itera- tions performed
IDN		PARRAY	downstream node number for coolant- channel pressure calculations
IDNS		PARRAY	downstream station number for coolant- channel pressure calculations
IDX		PARRAY	upstream node number for coolant- channel pressure calculations
IEND		GETIN	last point in CHANL array occupied by data for given slice
IFCP		FLOWs	indicator used in locating pressure-side film-cooling holes
IFCS		FLOWs	indicator used in locating suction-side film-cooling holes
IFILM	TCO		input indicator for film cooling
IFLU		PREP	coolant-channel node number
IFNL		PARRAY	number of coolant-channel nodes
IFNL		TCOEF	total number of stations, minus 1
IFSPLT		FLOWs	indicates in which direction film-cooling air flows from stagnation station
IGG(80)		HCOOL	array containing node numbers at which ratio of coolant crossflow to impingement-jet flow is out of Kercher-Tabakoff correlation range

Variable	Common	Subroutine	Definition
IGGC		HCOOL	counts number of entries in IGG array
IHC(80)	TCO		indicates type of coolant-side heat transfer at each station for given slice
IHCT		GETIN	input value of IHC for given station
IHUB	TCO		indicates type of boundary to be used at hub end of blade
II		HCOOL	coolant-channel node number
IIHCTZ		GETIN	locates IHC array in overall array INDCHN
ILEAD		HCOOL	last station in range of leading-edge impingement correlation
IMS		FLOWS	location of film-cooling hole preceding current station
INDCHN(2000)	SPECL		array for storing integer input data
INEDIT		GETIN INPRT	input variable to control listing of input data
INN		PREP	location of IHC data in INDCHN array
INSTRT		GETIN	starting point for storage of integer data in INDCHN array for given slice
INUM		NTTACT	number of stations on each side of blade
IN1		PREP	location of end of group of single-valued variables in INDCHN array
IPILOT	SPECL		input indicator to control plotting options
IPRES		WROUT	pressure-side, outer-surface node number
IPRSMN		WROUT	location of minimum outer-surface temperature on pressure side
IPRSMX		WROUT	location of maximum outer-surface temperature on pressure side

Variable	Common	Subroutine	Definition
IRE(80)		HCOOL	list of station numbers at which impingement-jet Reynolds number is out of Kercher-Tabakoff correlation range
IRL		PARRAY	coolant node number
ISBLOK	TCO		starting point in CHANL array of data for given slice
ISEN		TARRAY	indicates which side of blade station IS is on
ISENS		FLOWS	indicates which side of blade a given station is on
ISENS		TARRAY	indicates which side of blade a given trailing-edge region station is on
ISLICE	TCO		current slice number
ISTA		GETIN	input station number
ISTAT		WROUT	outer-surface node number for given station
ISTATD		WROUT	outer-surface node number immediately downstream of ISTAT
ISTB		GETIN	input station number
ISTRRT		HCOOL	station at which use of Kercher-Tabakoff correlation begins
ISTRRT		TCOEF	first station in trailing-edge region
ISUCMN		WROUT	location of minimum outer-surface temperature on suction side
ISUCMX		WROUT	location of maximum outer-surface temperature on suction side
ISUCT		WROUT	suction-side station number
ISUP		TARRAY	adjacent station, upstream of current station, IS
ISYM(5)		PLOTMF	data array containing plotting symbol codes

Variable	Common	Subroutine	Definition
ITDHFZ		GETIN	locates film-cooling-hole data in CHANL array for given slice
ITDHJZ		GETIN	locates impingement-hole data in CHANL array for given slice
ITDLXZ		GETIN	locates node spacing data in CHANL array for given slice
ITDPZ		GETIN	locates pin-fin diameter data in CHANL array for given slice
ITHKZ		GETIN	locates wall and channel thickness data in CHANL array for given slice
ITIP	TCO		indicates type of boundary to be used at tip end of blade
ITRBG		INPRT	first station in trailing-edge region
ITRBG		WROUT	first station in trailing-edge region
ITREO		WROUT	last outside node at trailing edge
ITRRZ		GETIN	locates radial position data in CHANL array for given slice
ITSPZ		GETIN	locates pin-fin spacing data in CHANL array for given slice
ITXNZ		GETIN	locates impingement-hole spacing data in CHANL array for given slice
IUNITS	UNITS		indicates system of units used for input data
IUNSTB		FLSPLT	indicates whether flow split is stable or not
IUP		PARRAY	upstream node number for coolant-channel pressure calculations
IUPS		PARRAY	upstream station number for coolant-channel pressure calculations
IVARS(12)		PLOTMF	array of integer plotting controls
IVERGE		TCOEF	pressure iteration loop counter
		WROUT	

Variable	Common	Subroutine	Definition
IWR		GAUSS	control on debugging output of coefficient matrix
IWRITE	SPECL		input control on amount of printed output
IXAX		PLOTMF	logical variable with value .TRUE.
IYAX		PLOTMF	logical variable with value .FALSE.
I1		PREP	starting point in INDCHN array for integer data for given slice, also used as starting point for nodal data in CHANL array for given slice
I3		PREP	starting point in CHANL array for station data
JDIS		FLOWS	number of stations that impingement flow-split station is displaced from station 1
JHCAL		INPRT	indicates type of heat transfer at given station
JLSTM		THRCON	size of wall thermal conductivity tables
JNUMS		FLSPLT	indicates whether previous call to FLSPLT resulted in unstable flow split
JOUTRG		FLSPLT	indicates attempt to split more than 100 percent of flow to one side
JPIV		GAUSS	pivotal column in matrix to be reduced
JS		FLOWS	station number at impingement flow-split point
		HCOOL	
		PARRAY	
		TARRAY	
		TCOEF	
		WROUT	

Variable	Common	Subroutine	Definition
JSENS		FLOWS GETIN FLSPLT PARRAY TARRAY TCOEF	indicates which side of blade impinge- ment flow-split is on
JSGNCH		FLSPLT	indicates whether a sign change has occurred in EPSN
JSO(15)		TCOEF	array for saving converged-flow-split station number for each slice
JSOLDS(25)		FLSPLT	array to keep track of stations checked as flow-split stations
JTIMES		FLSPLT	indicates whether DELTAN is to be rough adjusted or fine tuned
K		GAUSS	total bandwidth of matrix to be solved
K		NTTACT	counter of number of overall coolant- flow iterations
KSIG		PLNUM	coolant-plenum iteration counter
L		TARRAY THRCON	midwall node number at given station
LCOOL		FLOWs HCFRCD HCOOL HCPINS TARRAY TCOEF WROUT	coolant-channel node number at given station
LCOOLP		HCPINS TARRAY	inner-surface node across coolant- channel from given station
LCUP		HCPINS TARRAY	coolant-channel node number upstream of given station
LCUPP		TARRAY	inner-surface node across coolant chan- nel and upstream of given station

Variable	Common	Subroutine	Definition
LCUPS		TARRAY	inner-surface node upstream of given station
LDN		TARRAY	midwall node number downstream of given station
LIN		FLows HCFRCD HCOOL HCPINS TARRAY	inner-surface node number at given station
LJ		INPRT TARRAY THRCON	node number at junction of outer coating and wall metal at given station
LOUT		TARRAY THRCON	wall outer-surface node number at given station
LSP		PREP	starting point in BCXP array of data for given slice
LSS		PREP	starting point in BCXS array of data for given slice
LUP		TARRAY	midwall node number upstream of given station
MACH1		PLNUM	indicator to keep track of step-size change
MD1	SPECL		indicator to control special condensed, on-line output
MD2	SPECL		indicator that job is complete and summary plots are to be produced
MD3	SPECL		plot counter
MNBC		INPRT	maximum of NBCS and NBCP
N		GAUSS	number of rows in matrix to be solved
NAG		PLNUM	indicator whether calculations have progressed beyond initial station

Variable	Common	Subroutine	Definition
NBCP	BOUND		input number of boundary condition points on pressure side
NBCS	BOUND		input number of boundary condition points on suction side
NBFRP		INPRT	number of pressure-side boundary condition points preceding data for given slice at given time
NBFRS		INPRT	number of suction-side boundary condition points preceding data for given slice at given time
NBLKSZ	TCO		size of data block in CHANL array for given slice
NCC		PLNUM	loop counter
NCHAR		NTTACT	number of characters in ALPH2 array
NCOOL		PLOTMF	coolant node number
NEND		HCOOL	end of region that uses leading-edge impingement correlation
NFC		FLOWS	number of station containing film-cooling holes
NFCSUP(80)	FLMCOL		array identifying node supplying film cooling to each downstream node
NFLUID(200)		INPRT	array of coolant-channel node numbers for each station
NFWD	TCO		number of stations in forward region
NG	GAAS		number of temperature entries in gas property table, GS
NGEO		GETIN	number of NAMELISTs /GEO/ to be read in
NINTRV		PLOTMF	number of temperature intervals in summary plots
NIT		TCOEF	counter of number of overall coolant-flow iterations

Variable	Common	Subroutine	Definition
NL		INPRT	output line counter
NLBLS		PLOTMF	number of points at which symbols are to be plotted
NMM		PLOTMF	midwall node number
NMW		NTTACT	outer-surface node number
NODM		WROUT	midwall node
NODOUT		GETIN	outer-surface node
NODSF		FLOWS GETIN INPRT PREP FLSPLT PARRAY TARRAY TCOEF	number of nodes in forward region
NODST		GETIN INPRT PREP PARRAY TARRAY TCOEF NTTACT	total number of nodes for given slice
NODSTM		INPRT	total number of nodes minus 4
NOS		FLows INPRT WROUT PLOTMF	outer-surface node number
NPRCP		INPRT	number of points in each pressure-side boundary condition array for times preceding current time
NPRCS		INPRT	number of points in each suction-side boundary condition array for times preceding current time

Variable	Common	Subroutine	Definition
NPRTP		INPRT	number of points in each pressure-side boundary condition array per time step
NPRTS		INPRT	number of points in each suction-side boundary condition array per time step
NPTS		PLOTMF	number of points on given plot
NROW		GAUSS	number of matrix row to be displayed by debug output
NSAVE		TCOEF	coolant node number just upstream of exit, location of TSAVE
NSLICE	TCO		current slice number
NSTA	TCO		number of stations per slice
NSTAPS		PLOTMF	number of stations on each side of blade
NSTNS		PLNUM	number of spanwise stations per slice in coolant plenum
NTBC		GETIN	number of entries in input BCTIME array
NTIMES		INPRT	number of entries in BCTIME array
NTTG		PREP	time step number
NTYTM		NTTACT	
NUMS		FLSPLT	counter to force at least four attempts at a good flow split
P(2,15,80)	TCO		pressure at each node, for two consecutive time steps, lbf/in ²
PAVG		FLOWs	coolant-channel static pressure, lbf/in ² , used in calculating impingement hole flow rates
PBAR		FLOWs	pressure used in calculating square of coolant-channel Mach number

Variable	Common	Subroutine	Definition
PCNVRG		TCOEF	pressure-difference convergence criterion
PD		FLows GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	Prandtl number
PDTOG		HCOOL	Prandtl number based on coolant-supply temperature
PEX(400)	BOUND		input array containing tables of static pressure at trailing-edge coolant exhaust, lbf/in ²
PEXC		INPRT	static pressure at trailing-edge coolant exhaust in SI units, kPa, for given slice
PEXIT(15)	TCO		static pressures at trailing-edge coolant exhaust for each slice at given time, lbf/in ²
PEXOLD(15)		TCOEF	saved value of exhaust static pressure, lbf/in ² , used in setting initial guess of pressure distribution for subsequent time step
PEXTT		PLNUM	total pressure at exit of coolant plenum for given slice, lbf/in ²
PG(80)	FLMCOL		array containing hot-gas-side static pressure, lbf/in ² , at each station
PI		HCOOL	constant, 3.14159
PIM	TCO		impingement-supply pressure, lbf/in ²
PIMOLD(15)		TCOEF	saved value of impingement-supply pressure, lbf/in ² , for each slice

Variable	Common	Subroutine	Definition
PIN(15)	RADL		coolant total pressure, lbf/in ² , at entrance to each slice
PINS	HCPINS TARRAY		number of pin fins at given station
PIVOT	GAUSS		main diagonal term of row of matrix being solved
PLEGN(5)	PLOTMF		alphabetic array to label pressure-data plots
PLTYME(2)	PLOTMF		alphabetic variable to print transient time on each plot
POLD(15, 80)	TCOEF		saved values of coolant-channel pressure, lbf/in ² , from previous iteration
PP	PLNUM		intermediate pressure term
PPLEN	NTTACT		impingement-supply pressure, lbf/in ²
PROD	HCOOL		intermediate calculation result
PROPS	GETIN		NAMELIST name
PSAV(5)	TCOEF		array to save last four values of pressure at flow-split station, used to check convergence
PTEMP	PLNUM		intermediate pressure term
PTEXIT	PLNUM		coolant-plenum total pressure, lbf/in ² : entrance value going into subroutine, exit value coming out
PTIN	NTTACT		coolant-supply pressure for a given time, lbf/in ²
PTIO(50)	BOUND		input array of coolant-supply pressure, kPa (lbf/in ²), as function of time, sec
PTIOC	INPRT		initial coolant-supply pressure in SI units, kPa
PTNOLD	NTTACT		previous value of PTIN, lbf/in ²
PT1	PLNUM		calculated coolant-plenum inlet total pressure, lbf/in ²

Variable	Common	Subroutine	Definition
PUMP(80)	TCO		term to account for coolant pumping due to wheel rotation
PUMTRM		PARRAY	term to account for coolant pumping due to wheel rotation
PXX		PLNUM	average static pressure, lbf/in ² , in coolant plenum for given slice
QG(80)	TCO		hot-gas heat flux to blade at each station, Btu/hr · ft ²
QHUB(80)	BOUND		heat flux conducted to blade wall from hub platform at each station, Btu/hr · ft ²
QHUBIN(400)	BOUND		input table of hub heat flux at each station as function of time, W/m ² (Btu/hr · ft ²)
QSNK(80)	TCO		term to account for heat removal from wall by film-cooling flow through wall
QTIP(80)	BOUND		heat flux from blade wall at tip for each station, Btu/hr · ft ²
QTIPIN(400)	BOUND		input table of tip heat flux at each station as function of time, W/m ² (Btu/hr · ft ²)
R	TCO		gas constant; value for air is build in, 53.35 ft-lbf/lbm · °R
RATIO		THRCOM	interpolating fraction
RCHRD		TARRAY	dimensionless ratio of time increment to chordwise length increment squared at each station
RCHRDM		TARRAY	maximum value of RCHRD for a given slice
RCVRY		TARRAY	recovery factor
RE(80)	PRPS		coolant-channel Reynolds number at each station

Variable	Common	Subroutine	Definition
REFC(80)	FLMCOL		film-cooling flow Reynolds number at each station
REJ(80)		HCOOL	impingement-jet Reynolds number at each station
REJOVR(80)		HCOOL	array to save values of impingement-jet Reynolds number that are out of range of correlation
REY		PLNUM	coolant-plenum Reynolds number based on hydraulic diameter
RHOBAR		TARRAY	mean density in coolant channel, lbm/in ³
RHOC	TRNSNT		input density of outer coating, kg/m ³ (lbm/ft ³)
RHOM	TRNSNT		input density of wall metal, kg/m ³ (lbm/ft ³)
RHOVG(400)	BOUND		input table of hot-gas-side, free-stream mass velocity at each station as function of time, kg/m ² · sec (lbm/ft ² · sec)
RHOVGA(80)	FLMCOL		hot-gas-side, free-stream mass velocity at each station for given slice, lbm/ft ² · sec
RI		GETIN	input value of radial location of coolant-plenum inlet for given slice, cm (in.)
RIN(15)	RADL		table of RI values for each slice, in.
RO		GETIN	input value of radial location of coolant-plenum exit for given slice, cm (in.)
ROINV C		HCOOL	intermediate term in impingement correlation, ft ³ /lbm
ROINV J		HCOOL	intermediate term in impingement correlation, ft ³ /lbm
ROOT		PARRAY	intermediate term in pressure calculations

Variable	Common	Subroutine	Definition
ROUT(15)	RADL		table of RO values for each slice, in.
RR(80)	TCO		mean radial location of each station for a given slice
RRP		PLNUM	radial location, in.
RTEMP		PLNUM	radial location, in.
RTNARR(2)		PLOTMF	array containing maximum and minimum values of plot variables
RTRNV		TARRAY	dimensionless ratio of time increment to through-the-wall length increment squared
RTRNVM		TARRAY	maximum value of RTRNV for given slice
S(15)	TCO		span of each slice, in.
SEGMTS		PLNUM	number of segments in coolant plenum for given slice
SIGB		PLNUM	dummy variable used in coolant-plenum calculations
SIGC		PLNUM	dummy variable used in coolant-plenum calculations
SIGMA(20)		PLNUM	coolant velocity distribution in coolant plenum
SLEGN(5)		PLOTMF	alphameric array to label suction-side plots
SLP		HCPINS	mean pin-fin length at given station, in.
SP(80)	PRPS		pin-fin spacing at each station, in.
SPAN	TCO		radial span of given slice, in.
SPANC		INPRT	radial span of given slice in SI units, cm
SPHTC	TRNSNT		input specific heat of outer coating, J/kg · K (Btu/lbm · °F)
SPHTM	TRNSNT		input specific heat of wall metal, J/kg · K (Btu/lbm · °F)

Variable	Common	Subroutine	Definition
ST		HCOOL	Stanton number calculated from user-supplied impingement correlation
STANMX		HCOOL	Stanton number calculated from leading-edge impingement correlation
SV(3)		PLNUM	array to save values of SIGC
SYMBL(10)		PLOTMF	array of integers to be used as plot symbols
SYMBOL		PLOTMF	particular entry from SYMBL array
T(2, 15, 400)	TCO		calculated temperature at each node for each slice for two time steps, °F
TABOVE		TARRAY	midwall temperature at given station in slice above current slice, °F
TAU(400)	TCO		array of thickness values, in.
TBAR		FLOWS	coolant temperature, °R, used to calculate Mach number
TBAR		HCPINS	ratio of temperature drops in pin fins, pressure-side wall temperature minus mid-coolant-channel temperature to suction-side wall temperature minus mid-coolant-channel temperature
TBAR		WROUT	mean outer-surface temperature for given slice, °F
TBARM		WROUT	mean midwall temperature, °F
TBELOW		TARRAY	midwall temperature at given station in slice below current slice, °F
TBULK		WROUT	overall blade bulk-metal temperature, °F
TC		THRCON	mean temperature of blade cladding material, °F
TCIN		NTTACT	coolant temperature, °F
TCOF(400, 30)	MATRIX		array of coefficients to be solved for temperature or pressure

Variable	Common	Subroutine	Definition
TDHF		GETIN	function of film-cooling-hole size and spacing
TDHJ		GETIN	hydraulic diameter of impingement hole
TDLX(5)		GETIN	array containing lengths from nodes at upstream station to corresponding nodes at current station
TDP		GETIN	input pin-fin diameter, cm (in.)
TEM		INPRT	coolant-inlet absolute temperature, K ($^{\circ}$ R)
TEPS	TRNSNT		factor used to define time mean properties
TERM		FLSPLT	adjustment to flow-split parameter
TG(80)	TCO		hot-gas reference temperature at each station for given slice, $^{\circ}$ R
THETA1		TARRAY	intermediate terms in nodal energy equations
THETA2			
THETA3			
THETA4			
THETA5			
THETA6			
THETA8			
THETA9			
THK(3)		GETIN	input array of thickness values, cm (in.)
THUB(80)	BOUND		specified temperature at each blade hub station for given time, $^{\circ}$ F
THUBIN(400)	BOUND		input table of hub temperatures at each station as function of time, K ($^{\circ}$ F)
TIKLE(30)		GETIN	blank alphabetic array used to initialize title array

Variable	Common	Subroutine	Definition
TIN(15)	RADL		inlet total temperature in coolant plenum for each slice, °F
TIP1		TARRAY	term in conduction equation to account for specified tip temperature
TIP3		TARRAY	term in conduction equation to account for specified tip heat flux
TITL		GETIN	NAMELIST name
TITLE(30)	SPECL		alphabetic array of title information from input
TLABL1(21)		PLOTMF	alphabetic arrays used to put input title on plots
TLABL2(9)			
TMAXP		PLOTMF	maximum pressure-side temperature to be plotted
TMAXS		PLOTMF	maximum suction-side temperature to be plotted
TMFRAC		PREP	time-interpolating function
TMINP		PLOTMF	minimum pressure-side temperature to be plotted
TMINS		PLOTMF	minimum suction-side temperature to be plotted
TMP		GASTBL	temperature used to determine gas prop- erties, °R
TMP1		GASTBL	temperature used to determine gas prop- erties, °F
TOTSPN		WROUT	total span of blade, in.
TP		PLNUM	intermediate temperature variable
TPLEN		NTTACT	mean impingement-plenum static tem- perature, °F
TPM(500)		PLOTMF	table of pressure-side, midwall temper- atures for summary plots, K (°F)
TPMAX		WROUT	maximum pressure-side, outer-surface temperature for given slice, K (°F)

Variable	Common	Subroutine	Definition
TPMIN		WROUT	minimum pressure-side, outer-surface temperature for given slice, K ($^{\circ}$ F)
TPO(500)		PLOTMF	table of pressure-side, outer-surface temperatures for summary plots, K ($^{\circ}$ F)
TREDGE		TARRAY	intermediate term in nodal energy equation in trailing-edge region
TREPS		PARRAY TARRAY	same as TEPS
TRR		GETIN	input mean radial location of given station, cm (in.)
TRTRM		PARRAY	intermediate transient term in pressure equations
TRTRMC		TARRAY	intermediate transient term involving outer coating
TRTRMG		TARRAY	intermediate transient term involving coolant
TRTRMJ		TARRAY	intermediate transient term for coolant channel at entrance to trailing edge
TSAVE		TCOEF	temperature in coolant channel, just upstream of exit
TSM(500)		PLOTMF	table of suction-side, midwall temperature for summary plots, K ($^{\circ}$ F)
TSMAX		WROUT	maximum suction-side, outer-surface temperature for given slice, K ($^{\circ}$ F)
TSMIN		WROUT	minimum suction-side, outer-surface temperature for given slice, K ($^{\circ}$ F)
TSO(500)		PLOTMF	table of suction-side, outer-surface temperatures for summary plots, K ($^{\circ}$ F)
TSP		GETIN	input pin-fin spacing, cm (in.)
TTEMP		PLNUM	variable to save temperature value, $^{\circ}$ R

Variable	Common	Subroutine	Definition
TTEXIT		PLNUM	coolant total temperature in plenum, $^{\circ}\text{F}$
TTIN		NTTACT	coolant-supply temperature, $^{\circ}\text{F}$, at entrance to coolant plenum for given time
TTIO(50)	BOUND		input table of coolant-supply temperature, $\text{K} (^{\circ}\text{F})$, as function of time
TTIP(80)	BOUND		table of blade tip temperature, $^{\circ}\text{F}$, for given time
TTIPIN(400)	BOUND		input table of blade tip temperature, $\text{K} (^{\circ}\text{F})$, as function of time
TTOTC(80)		TCOEF	coolant total temperature at each station for given slice, $^{\circ}\text{F}$
TTX		PLNUM	coolant total temperature at inlet to coolant plenum for given slice, $^{\circ}\text{R}$
TTYME		WROUT	current time in transient, sec
TT1(20)		PLNUM	total-temperature distribution, $^{\circ}\text{F}$, in coolant plenum
TW		THRCON	midwall temperature for evaluating thermal conductivity
TXN		GETIN	input spanwise spacing of impingement jets at given station, cm (in.)
TXX		PLNUM	average static temperature in coolant plenum for given slice, $^{\circ}\text{F}$
TYME	TRNSNT		time in transient calculations, sec
TYMMAX	TRNSNT		maximum time to which transient is carried, sec
T0G	TCO		impingement-jet temperature for given slice, $^{\circ}\text{R}$
T1		PLNUM	inlet total temperature in coolant plenum for given slice, $^{\circ}\text{R}$
UA(2)		INPRT	alphabetic array containing area units

Variable	Common	Subroutine	Definition
UL(2)		INPRT	alphameric array containing length units
V		TCOEF	factor to accelerate convergence of pressure iterations
VARIB(15)		PLOTMF	alphameric array containing some plot labels
VARS(12)		PLOTMF	array containing plotting controls
VDP		HCPINS	pin-fin diameter at given station, in.
VOLBAR		TARRAY	coolant-channel volume element at entrance to trailing-edge region, in ³
VSP		HCPINS	pin-fin spacing at given station, in.
V1		PLNUM	intermediate term in coolant-plenum calculations
W(15)	RADL		coolant flow-rate at entrance to each slice, lbm/hr
WC		HCOOL	absolute value of coolant flow-rate, lbm/sec
WCHK(80)	CHKHOL		alphameric variable used to indicate choked flow in impingement jets
WCHKDM	CHKHOL		alphameric variable used to indicate choked flow in holes dumping coolant to trailing-edge region
WCHOKE		PLNUM	coolant flow-rate at entrance to given slice, lbm/hr
WCR		FLOWs	critical flow-rate, lbm/sec
WCROS(2,15,80)	TCO		coolant-channel crossflow rate at each station for each slice for two time steps, lbm/sec
WDUMP	TCO		rate of coolant flow being dumped directly from plenum to trailing-edge region, lbm/sec
WFC(80)	TCO		film-cooling flow rate at each station for given slice, lbm/sec

Variable	Common	Subroutine	Definition
WFCDUM		FLOWs	intermediate variable in film-cooling flow rate calculation
WFCDUM		PARRAY	total film-cooling flow rate at given station, lbm/sec
WIM	TCO		total impingement-jet flow rate for given slice, lbm/sec
WJ(15,80)	TCO		impingement-jet flow rate for each station for each slice, lbm/sec
WPLEN	BOUND		input initial guess at total coolant flow, kg/hr (lbm/hr)
WPLENC		INPRT	total coolant flow in SI units, kg/hr
WPLENO		NTTACT	variable to save previous estimate of total coolant flow, lbm/hr
WS	RADL		rotor speed at given time, rpm
WSVST(50)	BOUND		input table of rotor speed, rpm, versus time, sec
WUSED		NTTACT	cumulative amount of coolant used, up to current slice, lbm/hr
WXCP		TARRAY	coolant flow-rate times specific-heat term
WXX		PLNUM	coolant flow-rate at entrance to given slice, lbm/hr
X(80)		TCOEF	coolant-channel node locations, in., used to set initial pressure distribution
XBAR		FLOWs	term in film-cooling effectiveness correlation
XCC		INPRT	coolant-channel distance from station 1 to given station, cm (in.)
XDUM		FLOWs	dummy variable used to save location of film-cooling hole

Variable	Common	Subroutine	Definition
XFC(80)	FLMCOL		distance from station with film-cooling holes to downstream stations without film-cooling holes, in.
XIS		INPRT	inside-wall surface distance from station 1 to given station
XJN		INPRT	distance along junction of cladding and wall metal from station 1 to given station, cm (in.)
XX(4)		PLNUM	pressure change in coolant plenum
XL		HCOOL	length of inner surface used in leading-edge impingement correlation, in.
XL(4)		PLNUM	temperature change in coolant plenum
XLABL(29)		PLOTMF	alphameric array of plot labels
XLABL2(15)		PLOTMF	alphameric array of plot labels
XLBL(20)		PLOTMF	array of x-coordinates to be plotted as slice numbers
XMM		INPRT	midwall distance from station 1 to given station, cm (in.)
XMU		FLOW S GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	viscosity, lbm/ft · hr
XMUC(80)	FLMCOL		coolant viscosity, lbm/ft · hr, at each station, evaluated at mean temperature between inner-wall surface and bulk coolant temperatures
XMUM		FLOW S	hot-gas viscosity, lbm/ft · hr
XMUTOG		HCOOL	coolant viscosity based on coolant-supply temperature, lbm/ft · hr

Variable	Common	Subroutine	Definition
XN(80)	TCO		spanwise spacing of impingement holes at each station, in.
XNN		PLNUM	factor for increasing number of stations in coolant plenum
XOD		PREP	ratio of impingement-hole spacing to hydraulic diameter, at given station
XOS		INPRT	wall outer-surface distance from station 1 to given station, cm (in.)
XOVERD		HCOOL	ratio of impingement-hole spacing to hydraulic diameter, at given station
XOVRL		HCPINS	location of zero temperature gradient in pin fins
XP		HCOOL	length of pressure-side inner-wall surface in leading-edge impingement region
XP(80)		PLOTMF	pressure-side, dimensionless distance along midwall plane from station 1 to each station
XP	PREP		distance of given pressure-side station from station 1, in.
XPF	PREP		interpolating fraction in BCXP table
XPL	PLOTMF		overall length along pressure-side, midwall plane, cm (in.)
XS	HCOOL		length of suction-side, inner-wall surface in leading-edge impingement region
XS(80)	PLOTMF		suction-side, dimensionless distance along midwall plane from station 1 to each station
XS	PREP		distance of given suction-side station from station 1, in.
XSF	PREP		interpolating function in BCXS table

Variable	Common	Subroutine	Definition
XSL		PLOTMF	overall length along suction-side, mid-wall plane, cm (in.)
XTEST		PLNUM	convergence test variable
XTOT		WROUT	overall outer-surface length around blade, in.
XTOTMD		WROUT	overall midwall length around blade, in.
XXN		PLNUM	number of spanwise stations per slice in coolant plenum
Y(320)		PLOTMF	array containing temperature values to be plotted
YCNVUU		TARRAY	indicates forced-convection heat transfer at last forward-region station on pressure side
YCONV		TARRAY	indicates forced-convection heat transfer at given station
YCONVU		TARRAY	indicates forced-convection heat transfer at station immediately upstream of given station
YFINS		TARRAY	indicates pin-fin heat transfer at given station
YFINSU		TARRAY	indicates pin-fin heat transfer at station immediately upstream of given station
YFNSUU		TARRAY	indicates pin-fin heat transfer at last forward-region station on pressure side
YIMP		TARRAY	indicates impingement heat transfer at given station
YIMPU		TARRAY	indicates impingement heat transfer at station immediately upstream of given station
YIMPUU		TARRAY	indicates impingement heat transfer at last forward-region station on pressure side

Variable	Common	Subroutine	Definition
YLABL(7)		PLOTMF	alphameric array for labeling plots
YLABL2(11)		PLOTMF	alphameric array for labeling plots
YLBL(20)		PLOTMF	array of coordinates of points to be plotted as slice numbers
YMAX		PLOTMF	maximum value of y-coordinates on plot
YMIN		PLOTMF	minimum value of y-coordinates on plot
YPLABL(10)		PLOTMF	alphameric array for labeling plots
YTEM(80)		PLOTMF	array to be plotted
ZED		PLNUM	coolant-plenum pressure-drop parameter for given slice
ZOVERD		HCOOL	ratio of coolant-channel width to impingement-hole hydraulic diameter
Z1(15)		PLNUM	coolant-plenum pressure-drop parameter for each slice
Z3		PLNUM	intermediate term involving coolant flow
Z4		PLNUM	intermediate term involving coolant flow

PROGRAM LISTING

----SOURCE.NTFACT---THIS IS THE MAIN PROGRAM. BLOCK DATA SUBPROGRAM
GASDAT MUST BE LOADED FIRST.

TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE

* * * * *

TACT1

COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000), NTTACT-0008
Z BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NTTACT-0009
Z BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80), NTTACT-0010
Z QHUBIN(400), OHUB(80), TTIPIN(400), TTIP(80), NTTACT-0011
Z QTIPIN(400), QTIP(80), RHOVG(400), PEX(400), NTTACT-0012
Z BCTIME(50), TTIO(50), PTIO(50), WPLEN, NTTACT-0013
Z WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP NTTACT-0014
Z NTTACT-0015

COMMON /FLMCOL/ RHOVGA(80), FG(80), XFC(80), FLMEFF(80), NTTACT-0016
Z XMUC(80), EMES(80), REFC(80), NFCSUP(80) NTTACT-0017

COMMON /GAAS/ GS(200), NG NTTACT-0018

COMMON /RDL/ APLN(15), DPLN(15), RIN(15), ROUT(15), NTTACT-0019
Z PIN(15), TIN(15), W(15), WS NTTACT-0020

COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NTTACT-0021
Z IPLOT, MD1, MD2, MD3, IADJIN, IWRITE NTTACT-0022

COMMON /TCO/ ADUMP, BTA, CD, CP, TOG,
Z GAM, PIM, R, SPAN, NTTACT-0023
Z WDUMP, WIM, AKC(15,80), AKW(15,80), NTTACT-0024
Z A(400), AJET(80), AM2(80), CNUM(80), NTTACT-0025
Z DH(80), DHF(80), DHJ(80), NTTACT-0026
Z DLX(400), FF(80), HC(80), HG(80), NTTACT-0027
Z P(2,15,80), PEXIT(15), PUMP(80), QG(80), NTTACT-0028
Z QSNK(80), RR(80), S(15), T(2,15,400), NTTACT-0029
Z TG(80), TAU(400), WPC(80), NTTACT-0030
Z WJ(15,80), WCROS(2,15,80), XN(80), NTTACT-0031
Z ICOR, IFILM, IHUB, ITIP, NTTACT-0032
Z ISBLOK, ISLICE, NBLKSZ, NSLICE, NTTACT-0033
Z NPWD, NSTA, IHC(80) NTTACT-0034

COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,
Z DLTYME, TYME, TEPS, TYMMAX NTTACT-0035

COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSE(2), CMSPL(2), NTTACT-0036
Z CTMPP(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2), NTTACT-0037
Z CVISC(2), CRHOVG(2), IUNITS NTTACT-0038

DIMENSION DP(80), SP(80), ALPH(12), ALPH2(4), CD1(200)

TTIO = TOTAL TEMPERATURE OF BLADE COOLING AIR AT INLET
WPLEN = ESTIMATE OF COOLANT FLOW RATE - USED AS FIRST GUESS
PTIO = TOTAL PRESSURE OF BLADE COOLING AIR AT INLET
PEX = EXTERNAL GAS STREAM STATIC PRESSURE AT TRAILING EDGE

DATA ALPH/' THI' , 'S JO' , 'B WA' , 'S ST' , 'ARTE' , 'D AT' ,
Z ' , ' , ' , ' , ' , ' , ' , ' / NTTACT-0049

DATA NCHAR/16/ NTTACT-0050

MD1 = 0 NTTACT-0051

MD2 = 0 NTTACT-0052

NTTACT-0053

NTTACT-0054

NTTACT-0055

NTTACT-0056

NTTACT-0057

NTTACT-0058

NTTACT-0059

NTTACT-0060

```

TYME = -1.0 NTTACT--0061
DLTYME = 0.0 NTTACT--0062
C NTTACT--0063
C TO GET AN ABBREVIATED OUTPUT OF MID-WALL TEMPERATURES AT THE TERMINAL NTTACT--0064
C FOR EACH SLICE, ENTER: NTTACT--0065
C AT TACT.50;SET TACT.MD1=1 NTTACT--0066
C NTTACT--0067
C MD3 = 0 NTTACT--0068
C K = 1 NTTACT--0069
C NTTACT--0070
C RECORD STARTING TIME, TO BE USED TO IDENTIFY MICROFILM PLOTS NTTACT--0071
C NTTACT--0072
C CALL TIME(NCHAR,ALPH2) NTTACT--0073
ALPH(7) = ALPH2(3) NTTACT--0074
ALPH(8) = ALPH2(4) NTTACT--0075
ALPH(10) = ALPH2(1) NTTACT--0076
ALPH(11) = ALPH2(2) NTTACT--0077
WRITE(6,425) (ALPH(I),I=1,12) NTTACT--0078
WRITE(8,425) (ALPH(I),I=1,12) NTTACT--0079
C NTTACT--0080
C READ IN DATA NTTACT--0081
C NTTACT--0082
C CALL GETIN(IWRITE,TYMMAX,WSVST,IADJIN) NTTACT--0083
C NTTACT--0084
C WRITE TITLE PAGE NTTACT--0085
C NTTACT--0086
C WRITE(6,400) NTTACT--0087
C WRITE(6,425) (ALPH(I),I=1,12) NTTACT--0088
C WRITE(6,430) (TITLE(I),I=1,30) NTTACT--0089
400 FORMAT(1H1,/////.50X,'***** OUTPUT *****',//) NTTACT--0090
425 FORMAT(/36X,12A4) NTTACT--0091
430 FORMAT(//1X,30A4) NTTACT--0092
C NTTACT--0093
C TTIN = TTIO(1) NTTACT--0094
C PTIN = PTIO(1) NTTACT--0095
440 WPLENO = WPLEN NTTACT--0096
PTNOLD = PTIN NTTACT--0097
PTIO(1) = PTIN NTTACT--0098
TYME = 0.0 NTTACT--0099
NTYM = 1 NTTACT--0100
IF (DLTYME.GT.0.) NTYM = TYMMAX/DLTYME + 1 NTTACT--0101
NODST = 5*NSTA NTTACT--0102
C NTTACT--0103
C C START MARCHING NTTACT--0104
C NTTACT--0105
C DO 1100 ITYM = 1,NTYM NTTACT--0106
ITYME = ITYM-1 NTTACT--0107
NTTG = ITYM NTTACT--0108
TYME = ITYME*DLTYME NTTACT--0109
IF (ITYM.EQ.1) TYME = -1. NTTACT--0110
C NTTACT--0111
C NTTACT--0112
C NTTACT--0113
C NTTACT--0114
C -- EVALUATE TIME DEPENDENT BOUNDARY CONDITIONS ----- NTTACT--0115
C - NTTACT--0116
C NTTACT--0117
C PTIN = PTIO(1) NTTACT--0118
IF (TYME.LT.0.0) GO TO 490 NTTACT--0119
C NTTACT--0120

```

```

C-- LOCATE COOLANT SUPPLY PRESSURE FOR TIME
C
      DO 450 I = 4,50,2
      PTIN = PTIO(I-3)
      IF (PTIO(I).LE.0.0) GO TO 460
      IPTIO = I-1
      IF (TYME.LE.PTIO(I).AND.TYME.GT.PTIO(I-2)) GO TO 455
450    CONTINUE
455    DENOM = PTIO(IPTIO+1)-PTIO(IPTIO-1)
           IF (DENOM.GT.0.) PTIN = PTIO(IPTIO-2) +
           Z          (PTIO(IPTIO)-PTIO(IPTIO-2))*(TYME-PTIO(IPTIO-1))/DENOM
460    CONTINUE
C
C-- LOCATE COOLANT SUPPLY TEMPERATURE FOR TYME
C
      DO 470 I = 4,50,2
      TTIN = TTIO(I-3)
      IF (TTIO(I).LE.0.0) GO TO 490
      ITTIO = I-1
      IF (TYME.LE.TTIO(I).AND.TYME.GT.TTIO(I-2)) GO TO 475
470    CONTINUE
475    DENOM = TTIO(ITTIO+1)-TTIO(ITTIO-1)
           IF (DENOM.GT.0.) TTIN = TTIO(ITTIO-2) +
           Z          (TTIO(ITTIO)-TTIO(ITTIO-2))*(TYME-TTIO(ITTIO-1))/DENOM
490    CONTINUE
C
C LOCATE THE VALUE OF THE WHEEL SPEED FOR THE CURRENT TIME.
C
      WS = WSVST(1)
      IF (TYME.LE.0.0) GO TO 530
      DO 510 I = 4,50,2
      WS = WSVST(I-3)
      IF (WSVST(I).LE.0.0) GO TO 530
      IWS = I-1
      IF (TYME.LE.WSVST(I).AND.TYME.GT.WSVST(I-2)) GO TO 520
510    CONTINUE
520    DENOM = WSVST(IWS+1) - WSVST(IWS-1)
           IF (DENOM.GT.0.0) WS = WSVST(IWS-2) +
           Z          (WSVST(IWS)-WSVST(IWS-2))*(TYME-WSVST(IWS-1))/DENOM
530    CONTINUE
C
C
C LOCATE THE VALUE FOR PEXIT, GAS STATIC PRESSURE AT EXIT OF BLADE,
C FOR THE CURRENT TIME AND ALL SLICES.
C
      IF (TYME.GT.0.0) GO TO 533
      DO 532 I = 1,NSLICE
      PEXIT(I) = PEX(I)
      IF (PEX(I).LE.0.0) PEXIT(I) = PEX(1)
532    CONTINUE
533    CONTINUE
      IF (BCTIME(2).LE.0.0) GO TO 545
      DO 535 I = 2,50
      IPEX = I-1
      IF (TYME.LE.BCTIME(I).AND.TYME.GT.BCTIME(I-1)) GO TO 540
535    CONTINUE
540    DENOM = BCTIME(IPEX+1) - BCTIME(IPEX)
           IF (DENOM.EQ.0.0) GO TO 545
           TYMFRC = (TYME - BCTIME(IPEX))/DENOM
           DO 542 I = 1,NSLICE

```

NTTACT--0121
 NTTACT--0122
 NTTACT--0123
 NTTACT--0124
 NTTACT--0125
 NTTACT--0126
 NTTACT--0127
 NTTACT--0128
 NTTACT--0129
 NTTACT--0130
 NTTACT--0131
 NTTACT--0132
 NTTACT--0133
 NTTACT--0134
 NTTACT--0135
 NTTACT--0136
 NTTACT--0137
 NTTACT--0138
 NTTACT--0139
 NTTACT--0140
 NTTACT--0141
 NTTACT--0142
 NTTACT--0143
 NTTACT--0144
 NTTACT--0145
 NTTACT--0146
 NTTACT--0147
 NTTACT--0148
 NTTACT--0149
 NTTACT--0150
 NTTACT--0151
 NTTACT--0152
 NTTACT--0153
 NTTACT--0154
 NTTACT--0155
 NTTACT--0156
 NTTACT--0157
 NTTACT--0158
 NTTACT--0159
 NTTACT--0160
 NTTACT--0161
 NTTACT--0162
 NTTACT--0163
 NTTACT--0164
 NTTACT--0165
 NTTACT--0166
 NTTACT--0167
 NTTACT--0168
 NTTACT--0169
 NTTACT--0170
 NTTACT--0171
 NTTACT--0172
 NTTACT--0173
 NTTACT--0174
 NTTACT--0175
 NTTACT--0176
 NTTACT--0177
 NTTACT--0178
 NTTACT--0179
 NTTACT--0180

```

      IC = (IPEX-1)*NSLICE + I
      PEXIT(I) = PEX(IC) + (PEX(IC+NSLICE)-PEX(IC))*TYMFRC
542    CONTINUE
      IF (IFILM.NE.2) GO TO 545
C
C-- SET INTERPOLATED VALUES OF FREE STREAM RHO*V FOR THIS TIME
C
      DO 543 I = 1,NSTA
      IRO = (IPEX-1)*NSTA + I
      IRN = IPEX*NSTA + I
543    RHOVGA(I) = RHOVG(IRO) + TYMFRC* (RHOVG(IRN)-RHOVG(IRO))
C
C
545    CONTINUE
C
C
C-- SET TIME INTERPOLATED VALUES OF QHUB OR THUB AND QTIP OR TTIP.
      IF (BCTIME(2).LE.0.0) GO TO 555
      DO 550 I = 1,NSTA
      IQO = (IPEX-1)*NSTA + I
      IQN = IPEX*NSTA + I
      IF (IHUB.EQ.1) THUB(I) = THUBIN(IQO) +
      Z          TYMFRC* (THUBIN(IQN)-THUBIN(IQO))
      IF (IHUB.EQ.3) QHUB(I) = QHUBIN(IQO) +
      Z          TYMFRC* (QHUBIN(IQN)-QHUBIN(IQO))
      IF (ITIP.EQ.1) TTIP(I) = TTIPIN(IQO) +
      Z          TYMFRC* (TTIPIN(IQN)-TTIPIN(IQO))
      IF (ITIP.EQ.3) QTIP(I) = QTIPIN(IQO) +
      Z          TYMFRC* (QTIPIN(IQN)-QTIPIN(IQO))
550    CONTINUE
      GO TO 565
C
555    CONTINUE
      DO 560 I = 1,NSTA
      IF (IHUB.EQ.1) THUB(I) = THUBIN(I)
      IF (IHUB.EQ.3) QHUB(I) = QHUBIN(I)
      IF (ITIP.EQ.1) TTIP(I) = TTIPIN(I)
      IF (ITIP.EQ.3) QTIP(I) = QTIPIN(I)
560    CONTINUE
C
565    TCIN = TTIN
      IF (ITYM.GT.1) WPLEN = WUSED*PTIN/PTNOLD
570    WPLENO = WPLEN
      PTNOLD = PTIN
      WUSED = 0.0
C
C   CALCULATE TEMPERATURE AND PRESSURES FOR EACH SLICE OF THE BLADE
C
      DO 1000 I = 1,NSLICE
      ISLICE = I
C
C   FIRST DETERMINE IMPINGEMENT PLENUM CONDITIONS
C
      CALL PLNUM(WPLEN,PPLEN,PTIN,TPLEN,TCIN)
      TOG = TPLEN + 460.
      PIM = PPLEN
      IF (IUNITS.EQ.1) GO TO 860
      WRITE(6,800) I,PIM,TOG
800    FORMAT(1H2//10X,100('*')//30X,'THE IMPINGEMENT PLENUM CONDITIONS'
      Z           ' FOR SLICE NO.',I2,' ARE: '//60X,'PIM = ',F7.2,
      NTTACT--0181
      NTTACT--0182
      NTTACT--0183
      NTTACT--0184
      NTTACT--0185
      NTTACT--0186
      NTTACT--0187
      NTTACT--0188
      NTTACT--0189
      NTTACT--0190
      NTTACT--0191
      NTTACT--0192
      NTTACT--0193
      NTTACT--0194
      NTTACT--0195
      NTTACT--0196
      NTTACT--0197
      NTTACT--0198
      NTTACT--0199
      NTTACT--0200
      NTTACT--0201
      NTTACT--0202
      NTTACT--0203
      NTTACT--0204
      NTTACT--0205
      NTTACT--0206
      NTTACT--0207
      NTTACT--0208
      NTTACT--0209
      NTTACT--0210
      NTTACT--0211
      NTTACT--0212
      NTTACT--0213
      NTTACT--0214
      NTTACT--0215
      NTTACT--0216
      NTTACT--0217
      NTTACT--0218
      NTTACT--0219
      NTTACT--0220
      NTTACT--0221
      NTTACT--0222
      NTTACT--0223
      NTTACT--0224
      NTTACT--0225
      NTTACT--0226
      NTTACT--0227
      NTTACT--0228
      NTTACT--0229
      NTTACT--0230
      NTTACT--0231
      NTTACT--0232
      NTTACT--0233
      NTTACT--0234
      NTTACT--0235
      NTTACT--0236
      NTTACT--0237
      NTTACT--0238
      NTTACT--0239
      NTTACT--0240

```

```

      Z      ' PSIA',//60X,'TOG = ',F7.2,' DEG. R',//10X,100(**)) NTTACT--0241
C
C      IF (I.EQ.1) WRITE(6,850) WPLEN NTTACT--0242
850  FORMAT(1H ,//30X,'CENTRAL PLENUM FLOW IS ',F6.1,' LBM/HR',//) NTTACT--0243
      GO TO 890 NTTACT--0244
C
C      CTOG = TOG/1.8 NTTACT--0245
860  CPM = PIM/CPRSR(1) NTTACT--0246
      WWRITE(6,870) I,CPM,CTOG NTTACT--0247
870  FORMAT(1H2//10X,100(**)//30X,'THE IMPINGEMENT PLENUM CONDITIONS' NTTACT--0248
      Z      ' FOR SLICE NO.',I2,' ARE: '//60X,'PIM = ',F7.2, NTTACT--0249
      Z      ' KPA, '//60X,'TOG = ',F7.2,' K    '//10X,100(**)) NTTACT--0250
      CWPLEN = WPLEN/CMSPL(1) NTTACT--0251
      IF (I.EQ.1) WRITE(6,880) CWPLEN NTTACT--0252
880  FORMAT(1H ,//30X,'CENTRAL PLENUM FLOW IS ',F6.1,' KG/HR',//) NTTACT--0253
C
900  CONTINUE NTTACT--0254
C
C      THEN COMPUTE TEMPERATURES AND PRESSURES NTTACT--0255
C
C      CALL PREP(I,NTTG) NTTACT--0256
C
C      CALL TCOEF(IWRITE,WS,K,IPILOT,ALPH2) NTTACT--0257
      IF (IPILOT.GT.0) CALL PLOTMF(ALPH2) NTTACT--0258
      IF (MD1.EQ.0) GO TO 975 NTTACT--0259
C
C THIS BLOCK PRINTS SPECIAL CONDENSED OUTPUT TO THE TERMINAL IF MD1 > 0 NTTACT--0260
C
      IC1 = 0 NTTACT--0261
      DO 955 II = 1,NSTA,2 NTTACT--0262
      IC1 = IC1 + 1 NTTACT--0263
      NMW = 5*II - 4 NTTACT--0264
955  CD1(IC1) = T(2,I,NMW) - 460. NTTACT--0265
      DO 960 II = 2,NSTA,2 NTTACT--0266
      IC1 = IC1 + 1 NTTACT--0267
      NMW = 5*II-4 NTTACT--0268
960  CD1(IC1) = T(2,I,NMW) - 460. NTTACT--0269
      WRITE(8,962) I,K NTTACT--0270
962  FORMAT(' BLADE SLICE ',I2,', OVERALL FLOW LOOP ',I2, NTTACT--0271
      Z      ', SURFACE TEMPERATURE, (F), STARTING AT LEADING EDGE') NTTACT--0272
      INUM = NSTA/2 + 1 NTTACT--0273
      WRITE(8,964) (CD1(II),II=1,INUM) NTTACT--0274
964  FORMAT(' PRESSURE SIDE'/10 (2X,F7.1)) NTTACT--0275
      INUM = INUM + 1 NTTACT--0276
      WRITE(8,966) CD1(1),(CD1(II),II=INUM,NSTA) NTTACT--0277
966  FORMAT('// SUCTION SIDE'/10 (2X,F7.1)) NTTACT--0278
C
C
975  CONTINUE NTTACT--0279
C
C      CHECK HOW MUCH PLENUM FLOW IS LEFT NTTACT--0280
C
      WUSED = WUSED + 3600.*WIM NTTACT--0281
      EXCESW = WPLEN - 3600.*WIM NTTACT--0282
      IF (EXCESW.GT.0..AND.I.LT.NSLICE) WPLEN = EXCESW NTTACT--0283
      IF (EXCESW.LT.0..AND.I.LT.NSLICE) WRITE(6,985) I NTTACT--0284
985  FORMAT(' RAN OUT OF MASS FLOW IN BRANCH NO. ',I2) NTTACT--0285
1000 CONTINUE NTTACT--0286
C
C

```

```

1010 IF (IUNITS.EQ.2) WRITE(6,1010) WUSED
      FORMAT(/30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'
      Z      ' STEP IS ',F6.1,' LBM/HR')
      CWUSED = WUSED/CMSFL(1)
      IF (IUNITS.EQ.1) WRITE(6,1011) CWUSED
1011 FORMAT(/30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'
      Z      ' STEP IS ',F6.1,' KG/HR')
C
      DO 1040 I = 1,NSLICE
      DO 1020 J = 1,NODST
1020 T(1,I,J) = T(2,I,J)
      DO 1040 J = 1,NSTA
1040 P(1,ISLICE,J) = P(2,ISLICE,J)
      IF (TYME.GT.0.0) GO TO 1100
C
C--> ADJUST COOLANT FLOW AND RECALCULATE TEMPERATURES, ETC. FOR STEADYNTTACT--0316
C STATE CASE OR TIME =0.0
C
      EXCESW = WPLENO - WUSED
      IF (ABS(EXCESW).LT..01*WPLENO) GO TO 1100
      IF (IADJIN.GT.0) GO TO 1050
C
C --> NORMAL ADJUSTMENT IS ON WPLEN
C
      WPLEN = WPLENO - .995*EXCESW
      K = K + 1
      PTIN = PTNOLD
      TCIN = TTIN
      GO TO 570
C
C ----> SPECIAL CASE, FOR IADJPT > 0, ADJUSTMENT IS ON PTIN
C
1050 PTIN = PEXIT(1) + (PTNOLD-PEXIT(1))*(WPLENO/WUSED)**1.6
      WPLEN = WPLENO
      TCIN = TTIN
      K = K + 1
      GO TO 570
C
1100 CONTINUE
C
C
      IF (IUNITS.EQ.1) GO TO 3860
3000 WRITE(6,3500) K,WPLEN
3500 FORMAT(/,20X,80(''),/,.23X,I2,' LOOP(S) ON INITIAL COOLANT FLOW'
      Z      ' WERE USED. FINAL VALUE IS ',F7.2,' LBM/HR')
      WFITE(6,3600) EXCESW
3600 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' LBM/HR',/
      Z      20X,80(''))
      WRITE(6,425) (ALPH(I),I=1,12)
C
      GO TO 3900
C
3860 CWPLEN = WPLENO/CMSFL(1)
      CEXCSW = EXCESW/CMSFL(1)
      WRITE(6,3870) K,CWPLEN
3870 FORMAT(/,20X,80(''),/,.23X,I2,' LOOP(S) ON INITIAL COOLANT FLOW'
      Z      ' WERE USED. FINAL VALUE IS ',F7.2,' KG/HR')
      WRITE(6,3880) CEXCSW
3880 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' KG/HR',/
      Z      20X,80(''))

```

NTTACT--0301
NTTACT--0302
NTTACT--0303
NTTACT--0304
NTTACT--0305
NTTACT--0306
NTTACT--0307
NTTACT--0308
NTTACT--0309
NTTACT--0310
NTTACT--0311
NTTACT--0312
NTTACT--0313
NTTACT--0314
NTTACT--0315
NTTACT--0316
NTTACT--0317
NTTACT--0318
NTTACT--0319
NTTACT--0320
NTTACT--0321
NTTACT--0322
NTTACT--0323
NTTACT--0324
NTTACT--0325
NTTACT--0326
NTTACT--0327
NTTACT--0328
NTTACT--0329
NTTACT--0330
NTTACT--0331
NTTACT--0332
NTTACT--0333
NTTACT--0334
NTTACT--0335
NTTACT--0336
NTTACT--0337
NTTACT--0338
NTTACT--0339
NTTACT--0340
NTTACT--0341
NTTACT--0342
NTTACT--0343
NTTACT--0344
NTTACT--0345
NTTACT--0346
NTTACT--0347
NTTACT--0348
NTTACT--0349
NTTACT--0350
NTTACT--0351
NTTACT--0352
NTTACT--0353
NTTACT--0354
NTTACT--0355
NTTACT--0356
NTTACT--0357
NTTACT--0358
NTTACT--0359
NTTACT--0360

```

      WRITE(6,425) (ALPH(I),I=1,12)          NTTACT-- 0361
C
3900  CONTINUE          NTTACT-- 0362
      MD2 = 1          NTTACT-- 0363
      IF (NSLICE.GT.1.AND.IPLOT.GT.0) CALL PLOTMF(ALPH2)
      WRITE(6,425) (ALPH(I),I=1,12)          NTTACT-- 0364
      STOP          NTTACT-- 0365
      END          NTTACT-- 0366
                                         NTTACT-- 0367
                                         NTTACT-- 0368

C----SOURCE.NFLOEST          NFLOEST 0369
      SUBROUTINE FLOWS (JS,DELTAN,ICHOKE,AMCHOK)          NFLOEST 0370
C                                         NFLOEST 0371
C=====NFLOEST 0372
C                                         NFLOEST 0373
C- SOURCE.NFLOEST--          NFLOEST 0374
C                                         NFLOEST 0375
C THIS SUBROUTINE COMPUTES THE FOLLOWING---          NFLOEST 0376
C WJ, IMPINGEMENT JET FLOW RATES (LBM/SEC).          NFLOEST 0377
C WCROS, THE CHANNEL CROSSFLOWS (LBM/SEC).          NFLOEST 0378
C AM2, THE SQUARE OF THE CHANNEL MACH NUMBER.          NFLOEST 0379
C WDUMP, A FLOWRATE DUMPED DIRECTLY FROM CENTRAL PLENUM INTO TRAILING          NFLOEST 0380
C EDGE CHANNEL (LBM/SEC).          NFLOEST 0381
C WIM, THE TOTAL COOLANT FLOW USED FOR THIS SLICE (LBM/SEC).          NFLOEST 0382
C WFC, FILM COOLING HOLE FLOW RATES (LBM/SEC).          NFLOEST 0383
C FF, THE CHANNEL FRICTION FACTOR, EITHER FOR LAMINAR, TURBULENT, OR          NFLOEST 0384
C A PIN FIN ARRAY.          NFLOEST 0385
C PLMEFF, A FILM COOLING EFFECTIVENESS.          NFLOEST 0386
C                                         NFLOEST 0387
C=====NFLOEST 0388
C
      COMMON /CHKHOL/ WCHK(80), WCHKDM          NFLOEST 0389
C                                         NFLOEST 0390
C                                         NFLOEST 0391
      COMMON /FLMCOL/ RHOVGA(80), PG(80), XPC(80), PLMEFF(80),          NFLOEST 0392
Z           XMUC(80), EMES(80), REPC(80), NPCSUP(80)          NFLOEST 0393
C                                         NFLOEST 0394
      COMMON /FRIC/ ALPHA, BETA, DELTA, EPS          NFLOEST 0395
C                                         NFLOEST 0396
      COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),          NFLOEST 0397
Z           CPC(80), GAMC(80), DUMR1(80), DUMR2(80)          NFLOEST 0398
C                                         NFLOEST 0399
      COMMON /TCO/ ADUMP, BTA, CD, CP,          NFLOEST 0400
Z           GAM, PIM, R, SPAN, TOG,          NFLOEST 0401
Z           WDUMP, WIM, AKC(15,80), AKW(15,80),          NFLOEST 0402
Z           A(400), AJET(80), AM2(80), CNUM(80),          NFLOEST 0403
Z           DH(80), DHF(80), DHJ(80),          NFLOEST 0404
Z           DLX(400), FF(80), HC(80), HG(80),          NFLOEST 0405
Z           P(2,15,80),PEXIT(15), PUMP(80), QG(80),          NFLOEST 0406
Z           QSNK(80), RR(80), S(15), T(2,15,400),          NFLOEST 0407
Z           TG(80), TAU(400), WFC(80),          NFLOEST 0408
Z           WJ(15,80), WCROS(2,15,80), XN(80),          NFLOEST 0409
Z           ICOR, IFILM, IHUB, ITIP,          NFLOEST 0410
Z           ISBLOK, ISLICE, NBLKSZ, NSLICE,          NFLOEST 0411
Z           NPWD, NSTA, IHC(80)          NFLOEST 0412
C                                         NFLOEST 0413
      COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,          NFLOEST 0414
Z           DLTYME, TYME, TEPS, TYMMAX          NFLOEST 0415
C                                         NFLOEST 0416
      DATA CHKD/'0'/, UNCHKD/' '/          NFLOEST 0417
C                                         NFLOEST 0418
      DIMENSION DELTAN(15)          NFLOEST 0419
C                                         NFLOEST 0420

```

C FOLLOWING VARIABLES NEEDED TO CALCULATE FILM COOLING EFFECTIVENESS.
C THEY ARE TRANSMITTED THROUGH COMMON FLMCOL
C WHERE-

C PG IS GAS SIDE STATIC PRESSURE DISTRIBUTION; XFC IS THE
C DISTANCE A STATION IS DOWNSTREAM FROM THE NEAREST
C ROW OF FILM COOLING HOLES, (IN); FLMEFF IS THE CALCULATED
C FILM EFFECTIVENESS AT EACH STATION;
C XMUC IS COOLANT VISCOSITY BASED ON LOCAL COOLANT TEMPERATURE;
C EMES IS THE PRODUCT M*S, WHERE M IS THE BLOWING
C RATIO, AND S IS AN EQUIVALENT SLOT WIDTH; REFC IS THE FILM
C REYNOLDS NUMBER, BASED ON S; AND NFCSUP IDENTIFIES
C THE STATION NUMBER SUPPLYING FILM COOLING TO DOWNSTREAM STATIONS.

100 CONTINUE

C INITIALIZE HOLE CHOKING INDICATOR TO UNCHOKED
DO 101 I = 1, NSTA
101 WCHK(I) = UNCHKD
ICHOKE = 0
N = NSTA-1

C N = NODE NUMBER OF LAST FLOW CHANNEL NODE, AT EXIT OF TRAILING EDGE

C CALCULATE IMPINGEMENT FLOWS, AND FORWARD REGION FILM COOLING FLOWS

C

TMP = TOG
CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
GAMO = GAM
CPO = CP
NODSF = 5*NFWD

C

PAVG = P(2,ISLICE,1)
WJ(ISLICE,1) = 0.0
IF (PAVG.GT.PIM) GO TO 120
WCR=CD*PAVG*AJET(1)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
Z (PIM/PAVG)**((GAMO-1.0)/GAMO)
WJ(ISLICE,1)=PAVG/(R*TOG)*AJET(1)*CD*
Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
1 *(PIM/PAVG)**((GAMO-1.0)/GAMO)
WFC(1) = 0.0
IF (P(2,ISLICE,1).GT.PG(1)) WFC(1) = CD*.25*3.1415926*(DHF(1)**2)
Z *SQRT(32.2*p(2,ISLICE,1)*(P(2,ISLICE,1)-PG(1))/(R*T(2,ISLICE,5)))
IF(WCR.LT.WJ(ISLICE,1)) WCHK(1) = CHKD
IF(WCR.LT.WJ(ISLICE,1)) WJ(ISLICE,1)=WCR

C

120 PAVG = P(2,ISLICE,2)
WJ(ISLICE,2) = 0.0
IF (PAVG.GT.PIM) GO TO 130
WCR=CD*PAVG*AJET(2)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*
Z (PIM/PAVG)**((GAMO-1.0)/GAMO)
WJ(ISLICE,2)=PAVG/(R*TOG)*AJET(2)*CD*
Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
Z *(PIM/PAVG)**((GAMO-1.0)/GAMO)
WFC(2) = 0.0
IF (P(2,ISLICE,2).GT.PG(2)) WFC(2) = CD*.25*3.1415926*(DHF(2)**2)
Z *SQRT(32.2*p(2,ISLICE,2)*(P(2,ISLICE,2)-PG(2))/
Z (R*T(2,ISLICE,10)))
IF(WCR.LT.WJ(ISLICE,2)) WCHK(2) = CHKD
IF(WCR.LT.WJ(ISLICE,2)) WJ(ISLICE,2)=WCR

C

NFLOEST 0421
NFLOEST 0422
NFLOEST 0423
NFLOEST 0424
NFLOEST 0425
NFLOEST 0426
NFLOEST 0427
NFLOEST 0428
NFLOEST 0429
NFLOEST 0430
NFLOEST 0431
NFLOEST 0432
NFLOEST 0433
NFLOEST 0434
NFLOEST 0435
NFLOEST 0436
NFLOEST 0437
NFLOEST 0438
NFLOEST 0439
NFLOEST 0440
NFLOEST 0441
NFLOEST 0442
NFLOEST 0443
NFLOEST 0444
NFLOEST 0445
NFLOEST 0446
NFLOEST 0447
NFLOEST 0448
NFLOEST 0449
NFLOEST 0450
NFLOEST 0451
NFLOEST 0452
NFLOEST 0453
NFLOEST 0454
NFLOEST 0455
NFLOEST 0456
NFLOEST 0457
NFLOEST 0458
NFLOEST 0459
NFLOEST 0460
NFLOEST 0461
NFLOEST 0462
NFLOEST 0463
NFLOEST 0464
NFLOEST 0465
NFLOEST 0466
NFLOEST 0467
NFLOEST 0468
NFLOEST 0469
NFLOEST 0470
NFLOEST 0471
NFLOEST 0472
NFLOEST 0473
NFLOEST 0474
NFLOEST 0475
NFLOEST 0476
NFLOEST 0477
NFLOEST 0478
NFLOEST 0479
NFLOEST 0480

```

130 PAVG = P(2,ISLICE,3) NPLOEST 0481
WJ(ISLICE,3) = 0.0 NPLOEST 0482
IF (PAVG.GT.PIM) GO TO 140 NPLOEST 0483
WCR=CD*PAVG*AJET(3)/(R*T0G)*SQRT(64.4*GAMO*R*T0G/(GAMO+1.))* NPLOEST 0484
Z (PIM/PAVG)**((GAMO-1.0)/GAMO) NPLOEST 0485
WJ(ISLICE,3)=PAVG/(R*T0G)*AJET(3)*CD* NPLOEST 0486
Z SQRT(64.4*GAMO*R*T0G/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO))) NPLOEST 0487
Z *(PIM/PAVG)**((GAMO-1.0)/GAMO) NPLOEST 0488
WFC(3) = 0.0 NPLOEST 0489
IF (P(2,ISLICE,3).GT.PG(3)) WFC(3) = CD*.25*3.1415926*(DHF(3)**2) NPLOEST 0490
Z *SQRT(32.2*p(2,ISLICE,3)*(P(2,ISLICE,3)-PG(3))/ NPLOEST 0491
Z (R*T(2,ISLICE,15))) NPLOEST 0492
IF(WCR.LT.WJ(ISLICE,3)) WCHK(3) = CHKD NPLOEST 0493
IF(WCR.LT.WJ(ISLICE,3)) WJ(ISLICE,3)=WCR NPLOEST 0494
140 CONTINUE NPLOEST 0495
C NPLOEST 0496
DO 150 I = 4,NFWD NPLOEST 0497
PAVG = P(2,ISLICE,I) NPLOEST 0498
WJ(ISLICE,I) = 0.0 NPLOEST 0499
IF (PAVG.GT.PIM) GO TO 150 NPLOEST 0500
WCR=CD*PAVG*AJET(I)/(R*T0G)*SQRT(64.4*GAMO*R*T0G/(GAMO+1.))* NPLOEST 0501
Z (PIM/PAVG)**((GAMO-1.0)/GAMO) NPLOEST 0502
WJ(ISLICE,I)=PAVG/(R*T0G)*AJET(I)*CD* NPLOEST 0503
Z SQRT(64.4*GAMO*R*T0G/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO))) NPLOEST 0504
Z *(PIM/PAVG)**((GAMO-1.0)/GAMO) NPLOEST 0505
WFC(I) = 0.0 NPLOEST 0506
IF (P(2,ISLICE,I).GT.PG(I)) WFC(I) = CD*.25*3.1415926*(DHF(I)**2) NPLOEST 0507
Z *SQRT(32.2*p(2,ISLICE,I)*(P(2,ISLICE,I)-PG(I))/ NPLOEST 0508
Z (R*T(2,ISLICE,5*I))) NPLOEST 0509
IF(WCR.LT.WJ(ISLICE,I)) WCHK(I) = CHKD NPLOEST 0510
IF(WCR.LT.WJ(ISLICE,I)) WJ(ISLICE,I)=WCR NPLOEST 0511
150 CONTINUE NPLOEST 0512
C NPLOEST 0513
C CALCULATE CROSSFLOW RATE AT EACH STATION NPLOEST 0514
C NPLOEST 0515
200 WCROS(2,ISLICE,JS) = 0.0 NPLOEST 0516
JDIS = JS/2 NPLOEST 0517
C**** JDIS IS THE DISPLACEMENT OF THE FLOW SPLIT STATION FROM STATION 1 NPLOEST 0518
JSENS = JS - 2*JDIS NPLOEST 0519
C***** JSENS = 0, STATION IS ON SUCTION SIDE (EVEN); NPLOEST 0520
C = 1,STATION IS ON PRESSURE SIDE (ODD) NPLOEST 0521
JP = JS + 2 NPLOEST 0522
JM = JS - 2 NPLOEST 0523
IF (JM.LT.1) JM = 1 NPLOEST 0524
JFIN = JS + 1 NPLOEST 0525
JSTART = JS + 3 NPLOEST 0526
IF (JS.EQ.1) GO TO 220 NPLOEST 0527
IF (JSENS.EQ.1) GO TO 230 NPLOEST 0528
IF (JSENS.EQ.0) GO TO 270 NPLOEST 0529
NPLOEST 0530
C NPLOEST 0531
220 CONTINUE NPLOEST 0532
C**** WCROS AT A GIVEN STATION IS THE CROSSFLOW AT UPSTREAM STATION NPLOEST 0533
C PLUS IMPINGEMENT JET NPLOEST 0533
C FLOW FROM UPSTREAM STATION MINUS ANY FILM COOLING NPLOEST 0534
C FLOW AT THIS STATION. NPLOEST 0535
C NPLOEST 0536
C** THIS BLOCK IS EXECUTED IF FLOW SPLIT OCCURS AT STATION 1. ISTART=5 NPLOEST 0537
WCROS(2,ISLICE,2) = DELTAN(ISLICE)*(WJ(ISLICE,1)-WFC(1)) - WFC(2) NPLOEST 0538
WCROS(2,ISLICE,3)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,1)-WFC(1))-WFC(3) NPLOEST 0539
GO TO 320 NPLOEST 0540

```

```

C
230  CONTINUE
C***** THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON
C      THE PRESSURE SIDE.  JSSENS=1
      WCROS(2,ISLICE,JP)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,JS)-WPC(JS))
      Z-WPC(JP)
      WCROS(2,ISLICE,JM)=DELTAN(ISLICE)*(WJ(ISLICE,JS)-WPC(JS))-WPC(JM)
      IF (JM.EQ.1) GO TO 250
      IRNG = JDIS - 1
      DO 240 I = 1,IRNG
      IBK = (JM - 2*I)
      IBKP = IBK + 2
240   WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP)
      Z - WPC(IBK)
250   JNOD = JFIN
      DO 260 I = 2,JNOD,2
      IM8 = I-2
      IF (IM8.EQ.0) IM8 = 1
260   WCROS(2,ISLICE,I) = WCROS(2,ISLICE,IM8) + WJ(ISLICE,IM8)-WPC(I)
      GO TO 320
C
270   CONTINUE
C***** THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON
C      SUCTION SIDE.  JSSENS=0
      WCROS(2,ISLICE,JP)=DELTAN(ISLICE)*(WJ(ISLICE,JS)-WPC(JS))-WPC(JP)
      IF (JS.EQ.2) JM = 1
      WCROS(2,ISLICE,JM)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,JS)-WPC(JS))
      Z-WPC(JM)
      IF (JM.EQ.1) GO TO 300
      IF (JM.EQ.2) GO TO 290
      JM1 = JM-2
      DO 280 I = 2,JM1,2
      IBK = (JM-I)
      IBKP = IBK + 2
280   WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP)
      Z - WPC(IBK)
290   WCROS(2,ISLICE,1) = WCROS(2,ISLICE,2) + WJ(ISLICE,2) - WPC(1)
300   CONTINUE
C-----NOW UP THE PRESSURE SIDE
      JNOD = JFIN
      DO 310 I = 3,JNOD,2
310   WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2)-WPC(I)
C
320   CONTINUE
      ISTART = JSTART
      DO 330 I = ISTART,NFWD
      WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2) - WPC(I)
330   CONTINUE
C
C  CALCULATE CROSSFLOW RE & MACH NUMBER SQUARED, AND FILM COOLING RE,
C  FOR THE FORWARD REGION.
C
      DO 360 I = 1,NFWD
      LCOOL = 5*I
      LIN = LCOOL-1
      AMIN = A(LCOOL)
C
C  EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C  AND COOLANT BULK
C

```

NPLOEST 0541
 NPLOEST 0542
 NPLOEST 0543
 NPLOEST 0544
 NPLOEST 0545
 NPLOEST 0546
 NPLOEST 0547
 NPLOEST 0548
 NPLOEST 0549
 NPLOEST 0550
 NPLOEST 0551
 NPLOEST 0552
 NPLOEST 0553
 NPLOEST 0554
 NPLOEST 0555
 NPLOEST 0556
 NPLOEST 0557
 NPLOEST 0558
 NPLOEST 0559
 NPLOEST 0560
 NPLOEST 0561
 NPLOEST 0562
 NPLOEST 0563
 NPLOEST 0564
 NPLOEST 0565
 NPLOEST 0566
 NPLOEST 0567
 NPLOEST 0568
 NPLOEST 0569
 NPLOEST 0570
 NPLOEST 0571
 NPLOEST 0572
 NPLOEST 0573
 NPLOEST 0574
 NPLOEST 0575
 NPLOEST 0576
 NPLOEST 0577
 NPLOEST 0578
 NPLOEST 0579
 NPLOEST 0580
 NPLOEST 0581
 NPLOEST 0582
 NPLOEST 0583
 NPLOEST 0584
 NPLOEST 0585
 NPLOEST 0586
 NPLOEST 0587
 NPLOEST 0588
 NPLOEST 0589
 NPLOEST 0590
 NPLOEST 0591
 NPLOEST 0592
 NPLOEST 0593
 NPLOEST 0594
 NPLOEST 0595
 NPLOEST 0596
 NPLOEST 0597
 NPLOEST 0598
 NPLOEST 0599
 NPLOEST 0600

```

TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
XHUC(I) = XMU
CPC(I) = CP
GAMC(I) = GAM
C
RE(I) = 12.*3600.*ABS(WCROS(2,ISLICE,I))*DH(I)/(AMIN*XMU)
REPC(I) = 12.*3600.*WPC(I)/(S(ISLICE)*XMU)
IF (IHC(I).EQ.3) AMIN = A(LCOOL)*(SP(I)-DP(I))/SP(I)
W = WCROS(2,ISLICE,I)
PBAR = P(2,ISLICE,I)
TBAR = T(2,ISLICE,LCOOL)
335 AM2(I) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(I)/32.2
IF (AM2(I).LT.1.0) GO TO 340
ANCHOK = AM2(I)
ICHOKE = I
340 IF (IHC(I).NE.3) GO TO 350
AM2(I) = AM2(I)*(AMIN/A(LCOOL))**2
350 CONTINUE
360 CONTINUE
C
C CALCULATE FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION.
C
PAVG = .5*(P(2,ISLICE,NPWD) + P(2,ISLICE,NPWD-1))
IF (PAVG.GT.PIM) GO TO 370
WCR=CD*PAVG*ADUMP/(R*T0G)*SQRT(64.4*GAMO*R*T0G/(GAMO+1.))*Z
(PIM/PAVG)**((GAMO-1.0)/GAMO)
WDUMP=PAVG/(R*T0G)*ADUMP*CD*
Z SQRT(64.4*GAMO*R*T0G/(GAMO-1.)*(1.-(PAVG/PIM)**((GAMO-1.)/GAMO)))
Z *(PIM/PAVG)**((GAMO-1.0)/GAMO)
IF(WCR.LT.WDUMP) WCHKDM = CHKD
IF(WCR.LT.WDUMP) WDUMP=WCR
C
C ADD UP TOTAL FLOW FROM IMPINGEMENT PLENUM, WIM.
C
370 WIM = WDUMP
DO 380 I = 1,NSTA
380 WIM = WIM + WJ(ISLICE,I)
C
C TRAILING EDGE REGION, CALCULATE FILM COOLING FLOWS.
C
ISTRRT = NPWD+1
DO 400 I = ISTRRT,N,2
LCOOL = 5*I
WPCDUM = 0.0
390 IF (P(2,ISLICE,I).GT.PG(I)) WPCDUM = CD*.25*3.1415926*Z
SQRT(32.2*P(2,ISLICE,I)*(P(2,ISLICE,I)-PG(I))/Z
(R*T(2,ISLICE,LCOOL)))
WPC(I) = WPCDUM*(DHF(I)**2)
WPC(I+1) = WPCDUM*(DHF(I+1)**2)
400 CONTINUE
C
C TRAILING EDGE REGION, CALCULATE CROSSFLOW, RE, MACH NUMBER
C SQUARED, AND FILM COOLING RE.
C
WCROS(2,ISLICE,ISTRRT) = WCROS(2,ISLICE,NPWD-1) +
Z WCROS(2,ISLICE,NPWD) + WDUMP + WJ(ISLICE,NPWD-1)
Z + WJ(ISLICE,NPWD) - WPC(ISTRRT) - WPC(ISTRRT+1)
WCROS(2,ISLICE,ISTRRT+1) = WCROS(2,ISLICE,ISTRRT)
AMIN = A(NODSP+5)

```

```

C EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C AND COOLANT BULK
C
C TMP = (T(2,ISLICE,5*ISTRRT) + T(2,ISLICE,5*ISTRRT-1))/2.
C CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
C XMUC(ISTRRT) = XMU
C XMUC(ISTRRT+1) = XMU
C CPC(ISTRRT) = CP
C CPC(ISTRRT+1) = CP
C GAMC(ISTRRT) = GAM
C GAMC(ISTRRT+1) = GAM
C
C REPC(ISTRRT) = 12.*3600.*WFC(ISTRRT)/(S(ISLICE)*XMU)
C REPC(ISTRRT+1) = 12.*3600.*WFC(ISTRRT+1)/(S(ISLICE)*XMU)
C RE(ISTRRT) = 12.*3600.*ABS(WCROS(2,ISLICE,ISTRRT))*DH(ISTRRT)/
C             (AMIN*XMU)
C RE(ISTRRT+1) = RE(ISTRRT)
C IS = ISTRRT
C IF (IHC(IS).EQ.3) AMIN = A(NODSP+5)*(SP(IS)-DP(IS))/SP(IS)
C W = WCROS(2,ISLICE,ISTRRT)
C PBAR = P(2,ISLICE,ISTRRT)
C TBAR = T(2,ISLICE,NODSP+5)
405 AM2(ISTRRT) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(ISTRRT)/32.2
C IF (AM2(ISTRRT).LT.1.0) GO TO 410
C AMCHOK = AM2(ISTRRT)
C ICHOKE = ISTRRT
C 410 IF (IHC(IS).NE.3) GO TO 420
C AM2(ISTRRT) = AM2(ISTRRT)*(AMIN/A(NODSP+5))**2
420 AM2(ISTRRT+1) = AM2(ISTRRT)
C ISTRRT = ISTRRT + 2
C IS = NPWD + 1
DO 450 I = ISTRRT,N,2
C LCOOL = 5*I
C AMIN = A(LCOOL)
C IS = IS + 2
C IF (IHC(IS).EQ.3) AMIN = A(LCOOL)*(SP(IS)-DP(IS))/SP(IS)
C WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) - WFC(I) - WFC(I+1)
C WCROS(2,ISLICE,I+1) = WCROS(2,ISLICE,I)
C
C EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C AND COOLANT BULK
C
C TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LCOOL-1))/2.
C CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
C XMUC(I) = XMU
C XMUC(I+1) = XMU
C CPC(I) = CP
C CPC(I+1) = CP
C GAMC(I) = GAM
C GAMC(I+1) = GAM
C
C REPC(I) = 12.*3600.*WFC(I)/(S(ISLICE)*XMU)
C REPC(I+1) = 12.*3600.*WFC(I+1)/(S(ISLICE)*XMU)
C RE(I) = 12.*3600.*ABS(WCROS(2,ISLICE,I))*DH(I)/(A(LCOOL)*XMU)
C RE(I+1) = RE(I)
C W = WCROS(2,ISLICE,I)
C PBAR = P(2,ISLICE,I)
C TBAR = T(2,ISLICE,LCOOL)
425 AM2(I) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(I)/32.2

```

```

IF (AM2(I).LT.1.0) GO TO 430          NFLOEST 0721
AMCHOK = AM2(I)                      NFLOEST 0722
ICHOKE = I                           NFLOEST 0723
430  IF (IHC(IS).NE.3) GO TO 440      NFLOEST 0724
AM2(I) = AM2(I)*(AMIN/A(LCOOL))**2   NFLOEST 0725
440  AM2(I+1) = AM2(I)              NFLOEST 0726
450  CONTINUE                      NFLOEST 0727
C
C  CALCULATE COOLANT CHANNEL FRICTION FACTOR AT EACH STATION.
C
DO 560 I = 1,NSTA                  NFLOEST 0728
LCOOL = 5*I                         NFLOEST 0729
IF (WCROS(2,ISLICE,I).LE.0.0) GO TO 550  NFLOEST 0730
C
C  DETERMINE IF RE IS LAMINAR, TRANSITIONAL, OR TURBULENT
C  AND CALCULATE THE FRICTION FACTOR
C
IF (IHC(I).EQ.3) GO TO 540          NFLOEST 0731
IF (RE(I).GT.2300.) GO TO 510      NFLOEST 0732
500  FF(I) = DELTA*RE(I)**EPS       NFLOEST 0733
GO TO 560                           NFLOEST 0734
510  IF (RE(I).LT.4000.) GO TO 530  NFLOEST 0735
520  FF(I) = ALPHA*RE(I)**BETA     NFLOEST 0736
GO TO 560                           NFLOEST 0737
530  A1=DELTA*2300.**EPS           NFLOEST 0738
A2=ALPHA*4000.**BETA               NFLOEST 0739
FF(I) = (A2*(RE(I)-2300.)*A1*(4000.-RE(I)))/1700.  NFLOEST 0740
GO TO 560                           NFLOEST 0741
540  CONTINUE                      NFLOEST 0742
NFLOEST 0743
NFLOEST 0744
NFLOEST 0745
NFLOEST 0746
NFLOEST 0747
NFLOEST 0748
NFLOEST 0749
NFLOEST 0750
NFLOEST 0751
NFLOEST 0752
NFLOEST 0753
NFLOEST 0754
NFLOEST 0755
NFLOEST 0756
NFLOEST 0757
NFLOEST 0758
NFLOEST 0759
C
C  FOR A PIN FIN ARRAY:
FF(I) = 1.060*(RE(I)**(-.3301))
GO TO 560
550  CONTINUE                      NFLOEST 0760
FF(I) = 0.0
560  CONTINUE                      NFLOEST 0761
C
C
C  THE FOLLOWING BLOCK IS USED TO COMPUTE THE FILM COOLING EFFECTIVENESS
C  IF IFILM IS SET = 2
C
IF (IFILM.LT.2) GO TO 690          NFLOEST 0762
IF SPLT = 0                         NFLOEST 0763
DO 610 I = 1,NSTA                  NFLOEST 0764
610  XPC(I) = 0.0                   NFLOEST 0765
N = NSTA-1                          NFLOEST 0766
NFLOEST 0767
C LOCATE FILM COOLING HOLES AND SET UP THE XPC ARRAY
C
C---IFSPLT IS AN INDICATOR THAT TELLS WHICH SIDE OF THE BLADE STATION 1
C---          IS TO BE CONSIDERED A PART OF FOR FILM COOLING PURPOSES.
C---          = 0 IS THE DEFAULT, AND INDICATES SUCTION SIDE
C---          = 1 WILL INDICATE PRESSURE SIDE.
C FIRST, MARCH DOWN THE PRESSURE SIDE SEARCHING FOR FILM COOLING HOLES
IF (DHP(1).GT.0.0) NFC = IFSPLT    NFLOEST 0770
IF (NPC.EQ.0) NFC = NSTA + 1       NFLOEST 0771
XDUM = 0.0                          NFLOEST 0772
DO 615 I = 3,NSTA,2                NFLOEST 0773
NOS = 5*I - 4                      NFLOEST 0774
IF (I.GT.NFC) XPC(I) = XDUM + DLX(NOS)  NFLOEST 0775
NFLOEST 0776
NFLOEST 0777
NFLOEST 0778
NFLOEST 0779
NFLOEST 0780

```

```

IP (DHF(I).GT.0.0) GO TO 612          NFLOEST 0781
XDUM = XPC(I)                         NFLOEST 0782
GO TO 615                             NFLOEST 0783
612  NFC = I                           NFLOEST 0784
XDUM = 0.0                            NFLOEST 0785
615  CONTINUE                         NFLOEST 0786
C                                     NFLOEST 0787
C-- SUCTION SIDE                      NFLOEST 0788
C                                     NFLOEST 0789
IF (DHF(1).GT.0.0) NFC = 1 - IPSPLT   NPLOEST 0790
IF (NFC.EQ.0) NFC = NSTA + 1          NPLOEST 0791
XDUM = 0.0                            NPLOEST 0792
DO 625 I = 2,N,2                      NPLOEST 0793
NOS = 5*I - 4                        NPLOEST 0794
IF (I.GT.NFC) XPC(I) = XDUM + DLX(NOS) NFLOEST 0795
IF (DHF(1).GT.0.0) GO TO 622          NFLOEST 0796
XDUM = XPC(I)                         NFLOEST 0797
GO TO 625                             NFLOEST 0798
622  NFC = I                           NFLOEST 0799
XDUM = 0.0                            NFLOEST 0800
625  CONTINUE                         NFLOEST 0801
C INT. J. HT. & MASS TRANS., V8, 1965, PP 55-65
C     FLMEPP = 3.09*((X/(M*S))*(RE*MUC/MUG)**(-1/4) + 4.1)**(-.8)
C
IPCS = 0                             NPLOEST 0803
IPCP = 0                             NPLOEST 0804
IF (WPC(1).LE.0.0) GO TO 630         NFLOEST 0805
IPCS = 1 - IPSPLT                   NFLOEST 0806
IPCP = IPSPLT                       NFLOEST 0807
IF (RHOVGA(1).GT.0.0) EMES(1) = 144.*WPC(1)/(RHOVGA(1)*S(ISLICE)) NFLOEST 0808
630  CONTINUE                         NFLOEST 0809
DO 650 I=2,NSTA                     NFLOEST 0810
ISENS = I - 2*(I/2)                  NFLOEST 0811
IF (RHOVGA(I).GT.0.) EMES(I) = 144.*WPC(I)/(RHOVGA(I)*S(ISLICE)) NFLOEST 0812
FLMEPP(I) = 0.0                      NFLOEST 0813
IF (ISENS.EQ.0) GO TO 640           NFLOEST 0814
C PRESSURE SIDE SUPPLY HOLE LOCATIONS
IP (WPC(I).GT.0.0) IPCP = I          NFLOEST 0815
NPCSUP(I) = IPCP                    NFLOEST 0816
GO TO 650                           NFLOEST 0817
640  CONTINUE                         NFLOEST 0818
C SUCTION SIDE SUPPLY HOLE LOCATIONS
IP (WPC(I).GT.0.) IPCS = I          NFLOEST 0819
NPCSUP(I) = IPCS                    NFLOEST 0820
650  CONTINUE                         NFLOEST 0821
C
TMP = TG(1)                          NFLOEST 0822
CALL GASTBL(TMP,C,CPM,GAM,PD,R,XMUM) NFLOEST 0823
C
C FINALLY, CALCULATE THE EFFECTIVENESS
C
DO 680 I = 1,NSTA                  NFLOEST 0824
IMS = NPCSUP(I)                    NFLOEST 0825
IF (XPC(I).EQ.0.0.OR.EMES(IMS).EQ.0.0.OR.REPC(IMS).EQ.0.0) NFLOEST 0826
Z                                     GO TO 680
C3 = CPC(IMS)/CPM                  NFLOEST 0827
XBAR = (XPC(I)/EMES(IMS))*((REPC(IMS)*XMUC(I)/XMUM)**(-.25)) NFLOEST 0828
ETAPRM = 3.09*(XBAR+4.1)**(-.8)    NFLOEST 0829
FLMEPP(I) = C3*ETAPRM/(1.0 + (C3-1.0)*ETAPRM) NFLOEST 0830
IF (FLMEPP(I).GT.1.0) FLMEPP(I) = 1.0 NFLOEST 0831

```

```

680    CONTINUE          NFLOEST 0841
C
C
690    CONTINUE          NFLOEST 0842
    RETURN             NFLOEST 0843
    END               NFLOEST 0844
                      NFLOEST 0845
                      NFLOEST 0846

C----SOURCE.NFLSPLP
      SUBROUTINE FLSPLT(AJET,EPSN,ISLICE,NODSF,IDEKT,JS,DELTAN,ICONV)
      DIMENSION DELTAN(15), AJET(80), JSOLDS(25)

C   SOURCE.NFLSPLP---A SUBROUTINE TO SET THE STATION AT WHICH COOLING
C   AIR FLOW SPLITS BETWEEN THE SUCTION
C   AND THE PRESSURE SIDE FLOW CHANNELS.

C   INPUT TO FLSPLT IS THE PRESSURE MATCH PARAMETER, EPSN; THE NO. OF
C   NODES (NODSF) IN THE IMPINGEMENT REGION;
C   JS COMES IN AS THE CURRENT FLOW SPLIT STATION NO., AND IS
C   RETURNED AS THE NEW STATION IF A CHANGE IS NEEDED.
C   DELTAN COMES IN AS THE CURRENT FRACTION OF FLOW SPLIT TO
C   SUCTION SIDE FROM AN IMPINGEMENT
C   JET AT JS. IF A CHANGE IN JS IS NOT NEEDED, DELTAN IS
C   USED TO FINE TUNE THE SPLIT.
C   ICONV INDICATES IF CONVERGENCE IS COMPLETE.
C   = 0--NOT DONE; = 1--OK.

C   NPWD = NODSF/5
      IF (IUNSTB.EQ.1) GO TO 280
      IF (IDEKT.NE.1) GO TO 220
      JNUMS = 0
      IUNSTB = 0
      NUMS = 0
      JSGNCH=0
      JOUTRG=0
      DO 210 I = 1,25
210    JSOLDS(I) = 0
220    CONTINUE
      CRITR = .002
      ICONV = 0
      JSENS = JS - 2*(JS/2)
C***** (SUCTION - PRESSURE SIDE PRESSURES)/ SUCTION SIDE = EPSN
      IF (ABS(EPSN).LT.CRITR) GO TO 280
C
C*****IF EPSN < 0.0; NEED TO INCREASE FLOW TO PRESSURE SIDE
C***** EPSN > 0.0; NEED TO INCREASE FLOW TO SUCTION SIDE
C
      IF (JTIMES.EQ.0) GO TO 246
C
C***** JTGES = 0, THIS IS FIRST CHECK AT THIS STATION,
C           SO ROUGH ADJUST DELTAN;
C*****      1, HAVE BEEN HERE BEFORE, SO FINE TUNE DELTAN.
C
      IF ( JSGNCH.GT.0 ) GO TO 247
C
C***** JSGNCH = 0, THERE HAS NOT BEEN A PRIOR SIGN CHANGE IN EPSN;
C*****      = 1, THERE HAS BEEN A SIGN CHANGE BEFORE,
C           SO STAY AT THIS STATION
C
C
242    IF (EPSO/EPSN.LT.0.) JSGNCH = 1

```

```

C
247  CONTINUE
    IF (EPLAST/EPSN.GE.0) GO TO 243
    DELTAO = DELAST
    EPSO = EPLAST
243  CONTINUE
    IF (JSGNCH.EQ.0) GO TO 252
    IF (NUMS.GT.0) GO TO 248
    EPSMIN = ABS(EPSN)
    DLTAOP = DELTAN(ISLICE)
248  NUMS = NUMS + 1
    IF (ABS(EPSN).GT.EPSMIN) GO TO 249
    EPSMIN = ABS(EPSN)
    DLTAOP = DELTAN(ISLICE)
249  CONTINUE
    IF (NUMS.LT.4) GO TO 252
    IF (JNUMS.EQ.1) GO TO 250
    NUMS = 0
    JNUMS = 1
    DELTAN(ISLICE) = DELTAO
    JTGES = 0
    JSGNCH = 0
    JOUTRG = 0
    GO TO 290
250  CONTINUE
    DELAST = DELTAN(ISLICE)
    DELTAN(ISLICE) = DLTAOP
    IUNSTB = 1
    GO TO 290
C
C
246  JTGES = 1
    EPSO = EPSN
    DELTAO = DELTAN(ISLICE)
    IF (EPSO.GT.0.0) DELTAN(ISLICE) = (1.0+DELTAN(ISLICE))/2.0
    IF (EPSO.LT.0.0) DELTAN(ISLICE) = DELTAN(ISLICE)/2.
    IF (DELTAN(ISLICE).EQ.DELTAO) DELTAN(ISLICE) = DELTAN(ISLICE)+(.5-DELTAN(ISLICE))/5.
Z    GO TO 290
C
C
252  CONTINUE
    TERM = EPSN*(DELTAO-DELTAN(ISLICE))/(EPSO-EPSN)
    IF (TERM.EQ.0.) TERM = .05
    IF (JSGNCH.GT.0) GO TO 255
    DELTAO = DELTAN(ISLICE)
    EPSO = EPSN
255  CONTINUE
    DELAST = DELTAN(ISLICE)
    DELTAN(ISLICE) = DELTAN(ISLICE)-TERM
    IF (DELTAN(ISLICE).LT.1.0.AND.DELTAN(ISLICE).GT.0.0) GO TO 290
    IF (JOUTRG.GT.0) GO TO 258
    IF (DELTAN(ISLICE).LT.0.0) DELTAN(ISLICE) = .01
    IF (DELTAN(ISLICE).GT.1.0) DELTAN(ISLICE) = .99
    JOUTRG = 1
    GO TO 290
C
C
258  CONTINUE
    JOUTRG = 0

```

NFLSPLP 0901
NFLSPLP 0902
NFLSPLP 0903
NFLSPLP 0904
NFLSPLP 0905
NFLSPLP 0906
NFLSPLP 0907
NFLSPLP 0908
NFLSPLP 0909
NFLSPLP 0910
NFLSPLP 0911
NFLSPLP 0912
NFLSPLP 0913
NFLSPLP 0914
NFLSPLP 0915
NFLSPLP 0916
NFLSPLP 0917
NFLSPLP 0918
NFLSPLP 0919
NFLSPLP 0920
NFLSPLP 0921
NFLSPLP 0922
NFLSPLP 0923
NFLSPLP 0924
NFLSPLP 0925
NFLSPLP 0926
NFLSPLP 0927
NFLSPLP 0928
NFLSPLP 0929
NFLSPLP 0930
NFLSPLP 0931
NFLSPLP 0932
NFLSPLP 0933
NFLSPLP 0934
NFLSPLP 0935
NFLSPLP 0936
NFLSPLP 0937
NFLSPLP 0938
NFLSPLP 0939
NFLSPLP 0940
NFLSPLP 0941
NFLSPLP 0942
NFLSPLP 0943
NFLSPLP 0944
NFLSPLP 0945
NFLSPLP 0946
NFLSPLP 0947
NFLSPLP 0948
NFLSPLP 0949
NFLSPLP 0950
NFLSPLP 0951
NFLSPLP 0952
NFLSPLP 0953
NFLSPLP 0954
NFLSPLP 0955
NFLSPLP 0956
NFLSPLP 0957
NFLSPLP 0958
NFLSPLP 0959
NFLSPLP 0960

```

C
C
      JSGNCH = 0
      JTIMES = 0
      JSOLDS(JS) = 1
      IF (DELTAN(ISLICE).LT.1.) GO TO 265
C
C*** MOVE JS IN PRESSURE DIRECTION
      IF (JSENS.EQ.0) GO TO 262
261   JS = JS + 2
      IF (AJET(JS).LE.0.) GO TO 261
      GO TO 285
C
C
262   CONTINUE
      IF (JS.EQ.2) JS = 1
      IF (JS.GT.2) JS = JS - 2
      IF (AJET(JS).LE.0.) GO TO 262
      GO TO 285
C
C
265   CONTINUE
C*** MOVE JS IN SUCTION DIRECTION
      IF (JSENS.EQ.0) GO TO 267
      IF (JS.EQ.1) JS = 2
      IF (JS.GE.3) JS = JS - 2
      IF (AJET(JS).GT.0.) GO TO 285
      JSENS = JS - 2*(JS/2)
      GO TO 265
C
C
267   CONTINUE
      JS = JS + 2
      IF (AJET(JS).LE.0.) GO TO 267
      GO TO 285
C
C***** GET READY TO LEAVE SUBROUTINE
C
C THIS BLOCK IS EXECUTED IF CONVERGENCE WAS DETECTED
C
280   ICONV = 1
      JTIMES = 0
      IF (IUNSTB.EQ.1) WRITE(6,284) ISLICE,IDELT,JS,DELTAN(ISLICE)
      IF (IUNSTB.EQ.1) WRITE(8,284) ISLICE,IDELT,JS,DELTAN(ISLICE)
      IUNSTB = 0
      EPLAST = EPSN
      RETURN
284   FORMAT(1H2,40(' '),40('*')//' SLICE ',I2,', POOR FLOW SPLIT, ',
      Z     I3,' ITERATIONS, SPLIT AT STATION ',
      Z I2,', BEST SPLIT IS AT DELTA = ',F6.4)
C
C
C THIS BLOCK IS EXECUTED FOR AN ABNORMAL EXIT---PROGRAM IS TERMINATED
C
789   CONTINUE
      WRITE(6,792) DELTAN(ISLICE)
      WRITE(8,792) DELTAN(ISLICE)
      STOP
792   FORMAT(/5X,' ***** FLOW SPLIT CANNOT BE MADE AS SPECIFIED',
      Z           ' DELTA = ',F9.5)

```

```

C NPLSPLP 1021
C NPLSPLP 1022
285 IF ( JSOLDS (JS) .EQ. 1) GO TO 789 NPLSPLP 1023
DELTAN (ISLICE) = .50 NPLSPLP 1024
C NPLSPLP 1025
C NPLSPLP 1026
C THIS BLOCK IS THE USUAL EXIT AFTER ADJUSTING THE FLOW SPLIT NPLSPLP 1027
C NPLSPLP 1028
290 CONTINUE NPLSPLP 1029
EPLAST = EPSN NPLSPLP 1030
IF (JSGNCH.EQ.0) DELAST = DELTAO NPLSPLP 1031
IDELT = IDELT + 1 NPLSPLP 1032
RETURN NPLSPLP 1033
END NPLSPLP 1034

C----SOURCE.NGASDAT NGASDAT 1035
BLOCK DATA NGASDAT 1036
C NGASDAT 1037
C-- SOURCE.NGASDAT--- NGASDAT 1038
C NGASDAT 1039
COMMON /GAAS/ GS(200),NG NGASDAT 1040
DATA GS/620., 1160., 1700., 2240., 2780., 3320., NGASDAT 1041
Z .02564, .03580, .04548, .05467, .06435, .07475, NGASDAT 1042
Z .2511, .2681, .2814, .2939, .3070, .3214, NGASDAT 1043
Z .706, .706, .705, .703, .702, .699, NGASDAT 1044
Z .07233, .09458, .11369, .13063, .14683, .16256, NGASDAT 1045
Z 170*0.0/, NG /6/ NGASDAT 1046

C NGASDAT 1047
C---GS IS TABLE OF AIR PROPERTIES VS TEMPERATURE AT CONSTANT PRESSURE NGASDAT 1048
C--- PROPERTY VALUES ARE FROM POPEL & SVEHLA, TN D-7488, AT 20 ATM. NGASDAT 1049
C--- NG IS THE NUMBER OF TEMPERATURE ENTRIES IN THE TABLE NGASDAT 1050
C--- ENTRIES IN GS ARE: NGASDAT 1051
C--- 1ST NG ARE TEMPERATURE, (F) NGASDAT 1052
C--- 2ND NG ARE THERMAL CONDUCTIVITY, (BTU/(HR*FT*R)) NGASDAT 1053
C--- 3RD NG ARE SPECIFIC HEAT, (BTU/(LBM*R)) NGASDAT 1054
C--- 4TH NG ARE PRANDTL NUMBER NGASDAT 1055
C--- 5TH NG ARE VISCOSITY, (LBM/(FT*HR)) NGASDAT 1056
C NGASDAT 1057
END NGASDAT 1058
C----SOURCE.NGASTB NGASTB 1059
SUBROUTINE GASTBL(TMP,C,CP,GAM,PD,R,XMU) NGASTB 1060
C NGASTB 1061
C- SOURCE.NGASTB NGASTB 1062
C NGASTB 1063
C A SUBROUTINE TO LOOK UP GAS PROPERTIES IN AN INPUT TABLE (GS(200)) NGASTB 1064
C WHERE TMP = TEMPERATURE AT WHICH PROPERTIES ARE TO BE EVALUATED (R) NGASTB 1065
C C = GAS THERMAL CONDUCTIVITY (BTU/(HR*FT*R)) NGASTB 1066
C CP = GAS SPECIFIC HEAT (BTU/(LBM*R)) NGASTB 1067
C GAM = RATIO OF SPECIFIC HEATS NGASTB 1068
C PD = PRANDTL NUMBER NGASTB 1069
C R = SPECIFIC GAS CONSTANT (FT*LBF)/(LBM*R) NGASTB 1070
C XMU = VISCOSITY (LBM/(FT*HR)) NGASTB 1071
C NGASTB 1072
COMMON /GAAS/ GS(200),NG NGASTB 1073
DIMENSION AC(5) NGASTB 1074
C NGASTB 1075
TMP1=TMP - 460. NGASTB 1076
IF (TMP1.GT.GS(1)) GO TO 200 NGASTB 1077
100 AP1=0.0 NGASTB 1078
AP2=1.0 NGASTB 1079
I1=2 NGASTB 1080

```

```

I2=1          NGASTB 1081
GO TO 500    NGASTB 1082
200 DO 300 I=1,NG NGASTB 1083
      I1=I          NGASTB 1084
      IF(GS(I).GT.TMP1) GO TO 400 NGASTB 1085
300 CONTINUE   NGASTB 1086
      TMP1=GS(NG) NGASTB 1087
400 I2=I1-1    NGASTB 1088
      AP1=(TMP1-GS(I2))/(GS(I1)-GS(I2)) NGASTB 1089
      AP2=1.0-AP1 NGASTB 1090
500 DO 600 J=1,4 NGASTB 1091
      I1=I1+NG NGASTB 1092
      I2=I2+NG NGASTB 1093
600 AC(J)=AP1*GS(I1)+AP2*GS(I2) NGASTB 1094
      AC(5)=1.0/(1.0-R/(778.2*AC(2))) NGASTB 1095
      C=AC(1) NGASTB 1096
      CP=AC(2) NGASTB 1097
      PD=AC(3) NGASTB 1098
      XMU=AC(4) NGASTB 1099
      GAM=AC(5) NGASTB 1100
      RETURN NGASTB 1101
      END NGASTB 1102

```

```

C---- SOURCE.NGAUS          NGAUS 1103
      SUBROUTINE GAUSS(N,K)  NGAUS 1104
C                                         NGAUS 1105
C- SOURCE.NGAUS---           NGAUS 1106
C                                         NGAUS 1107
C---- GIVEN A COMPRESSED VERSION OF AN AUGMENTED, BAND MATRIX A NGAUS 1108
C---- WHERE K IS THE WIDTH OF THE BAND, N IS THE NUMBER OF ROWS (MAX 400) NGAUS 1109
C---- DIAGONAL ELEMENTS OF THE ORIGINAL MATRIX ARE STORED IN NGAUS 1110
C---- COLUMN ((K/2)+1) -- TCOP(I,((K/2)+1)) NGAUS 1111
C---- THE ORIGINAL RIGHT HAND SIDE IS IN COLUMN K+1--- TCOP(I,K+1) NGAUS 1112
C---- GAUSS ELIMINATION IS USED TO MAKE ALL ELEMENTS BELOW NGAUS 1113
C---- THE DIAGONAL ZERO. NGAUS 1114
C---- BACK-SUBSTITUTION IS USED TO COMPUTE THE X'S, WHICH ARE NGAUS 1115
C---- RETURNED IN TCOP(I,K+1) NGAUS 1116
C                                         NGAUS 1117
      REAL*8 TCOP          NGAUS 1118
      COMMON /MATRIX/ TCOP(400,30) NGAUS 1119
      IWR = 0               NGAUS 1120
      NROW = 0               NGAUS 1121
      IF (IWR.EQ.0) GO TO 63 NGAUS 1122
C                                         NGAUS 1123
C DEBUGGING OUTPUT:          NGAUS 1124
C                                         NGAUS 1125
C IWR CAN BE SET DYNAMICALLY IN ORDER TO GET DEBUG OUTPUT OF NGAUS 1126
C SELECTED ROWS OF THE MATRIX, BEFORE OR AFTER REDUCTION. NGAUS 1127
C                                         NGAUS 1128
      WRITE(6,57)          NGAUS 1129
57      FORMAT(' ENTER NUMBER OF ROW TO BE DISPLAYED. USE I3 FORMAT') NGAUS 1130
58      READ(7,59) NROW NGAUS 1131
59      FORMAT(I3)          NGAUS 1132
      IF (NROW.EQ.0) GO TO 63 NGAUS 1133
      KP = K+1             NGAUS 1134
      WRITE(6,60) NROW NGAUS 1135
      WRITE(6,61) (I,TCOP(NROW,I),I=1,KP) NGAUS 1136
60      FORMAT(/' TCOF MATRIX, ROW NO. ',I3) NGAUS 1137
61      FORMAT(5(' ,I3,'',D17.10,'')) NGAUS 1138
      WRITE(6,62)          NGAUS 1139
62      FORMAT(/' ENTER ANOTHER ROW NO. OR 000 TO CONTINUE PROCESSING') NGAUS 1140

```

```

GO TO 58                               NGAUS 1141
C                                         NGAUS 1142
C                                         NGAUS 1143
63  CONTINUE                           NGAUS 1144
65  JPIV = K/2 + 1                     NGAUS 1145
   N1 = N-1                            NGAUS 1146
   DO 100 I = 1,N1                      NGAUS 1147
   JS = I+1                            NGAUS 1148
   JP = I + K/2                         NGAUS 1149
   IF (JP.GT.N) JP = N                  NGAUS 1150
   PIVOT = TCOF(I,JPIV)                NGAUS 1151
   IF (PIVOT.EQ.0.0) GO TO 130          NGAUS 1152
   DO 90 J = JS,JP                      NGAUS 1153
   JR = JPIV-J+1                        NGAUS 1154
   IF (DABS(TCOF(J,JP)).LT.1.0D-30) GO TO 90
   FM = TCOF(J,JP)/PIVOT               NGAUS 1155
   TCOF(J,JP) = 0.0                      NGAUS 1156
   LS = JR + 1                          NGAUS 1157
   LF = LS + K/2                        NGAUS 1158
   IF (LF.LT.LS) GO TO 85              NGAUS 1159
   DO 80 L = LS,LF                      NGAUS 1160
   LR = L+JPIV+1-LS                    NGAUS 1161
   IF (LR.GT.K) GO TO 85              NGAUS 1162
80   TCOF(J,L) = TCOF(J,L) - FM*TCOF(I,LR)
85   TCOF(J,K+1) = TCOF(J,K+1) - FM*TCOF(I,K+1)
90   CONTINUE                           NGAUS 1163
100  CONTINUE                           NGAUS 1164
C                                         NGAUS 1165
C                                         NGAUS 1166
155  IF (IWR.EQ.0) GO TO 163           NGAUS 1167
C                                         NGAUS 1168
C DEBUGGING OUTPUT:                   NGAUS 1169
C                                         NGAUS 1170
158  WRITE(8,57)                       NGAUS 1171
   READ(7,59) NROW                     NGAUS 1172
   IF (NROW.EQ.0) GO TO 163            NGAUS 1173
   KP = K+1                            NGAUS 1174
   WRITE(8,60) NROW                     NGAUS 1175
   WRITE(8,61) (I,TCOF(NROW,I),I=1,KP)
   WRITE(8,62)                         NGAUS 1176
   GO TO 158                           NGAUS 1177
C                                         NGAUS 1178
C                                         NGAUS 1179
163  CONTINUE                           NGAUS 1180
   TCOF(N,K+1) = TCOF(N,K+1)/TCOF(N,JPIV)
   DO 120 I = 1,N1                      NGAUS 1181
   IIN = N-I                            NGAUS 1182
   JP = K/2                            NGAUS 1183
   SUM = TCOF(IIN,K+1)                 NGAUS 1184
   DO 115 J = 1,JP                      NGAUS 1185
   JP = J + JPIV                        NGAUS 1186
   IJ = J+IIN                          NGAUS 1187
   IF (IJ.GT.N) GO TO 117              NGAUS 1188
115  SUM = SUM - TCOF(IJ,K+1)*TCOF(IIN,JP)
117  CONTINUE                           NGAUS 1189
120  TCOF(IIN,K+1) = SUM/TCOF(IIN,JPIV)
125  CONTINUE                           NGAUS 1190
   RETURN                             NGAUS 1191
130  WRITE(7,135) I                   NGAUS 1192
135  FORMAT(/' DIAGONAL ELEMENT FOR ROW ',I2,' IS ZERO. NO ', NGAUS 1193
                                NGAUS 1194
                                NGAUS 1195
                                NGAUS 1196
                                NGAUS 1197
                                NGAUS 1198
                                NGAUS 1199
                                NGAUS 1200

```

Z	'FURTHER ATTEMPT TO SOLVE WILL BE MADE.'	NGAUS	1201
GO TO 125		NGAUS	1202
END		NGAUS	1203

C----	SOURCE.NGETINT	NGETINT	1204
	SUBROUTINE GETIN	NGETINT	1205
C		NGETINT	1206
C-	SOURCE.NGETINT----	NGETINT	1207
C		NGETINT	1208
Z	COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000), NGETINT 1209	BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NGETINT 1210	
Z	BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80), NGETINT 1211	QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NGETINT 1212	
Z	QTIPIN(400), QTIP(80), RHOVG(400), PEX(400), NGETINT 1213	BCTIME(50), TTIO(50), PTIO(50), WPLEN, NGETINT 1214	
Z	WSVST(50), AKTBL(20), AKWTBL(20), NBCS, NBCP NGETINT 1215	AKCTBL(20), NBCS, NBCP NGETINT 1216	
C	COMMON /FLMCOL/ RHOVG(80), PG(80), XPC(80), FLMEFF(80), NGETINT 1217	XMUC(80), EMES(80), REPC(80), NFCSUP(80) NGETINT 1218	
C	COMMON /IMPCOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7, NGETINT 1219	DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6 NGETINT 1220	
C	COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15), NGETINT 1221	PIN(15), TIN(15), W(15), WS NGETINT 1222	
C	COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NGETINT 1223	IPILOT, MD1, MD2, MD3, IADJIN, IWRITE NGETINT 1224	
C	COMMON /TCO/ ADUMP, BTA, CD, CP, NGETINT 1225	GAM, PIM, R, SPAN, TOG, NGETINT 1226	
Z	WDUMP, WIM, AKC(15,80), AKW(15,80), NGETINT 1227	A(400), AJET(80), AM2(80), CNUM(80), NGETINT 1228	
Z	DH(80), DHF(80), DHJ(80), NGETINT 1229	DLX(400), FF(80), HC(80), HG(80), NGETINT 1230	
Z	P(2,15,80), PEXIT(15), PUMP(80), OG(80), NGETINT 1231	QSNK(80), RR(80), S(15), T(2,15,400), NGETINT 1232	
Z	TG(80), TAU(400), WFC(80), XN(80), NGETINT 1233	WJ(15,80), WCROS(2,15,80), YH(80), NGETINT 1234	
Z	ICOR, IFILM, IHUB, ITIP, NGETINT 1235	ISBLOK, ISLICE, NBLKSZ, NSLICE, NGETINT 1236	
Z	NFWD, NSTA, IHC(80), NGETINT 1237	NSTA, NGETINT 1238	
C	COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM, NGETINT 1239	DLTYME, TYME, TEPS, TYMMAX, NGETINT 1240	
C	COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2), NGETINT 1241	CTMPF(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2), NGETINT 1242	
Z	CVISC(2), CRHOVG(2), IUNITS, NGETINT 1243	CVISC(2), CRHOVG(2), IUNITS, NGETINT 1244	
C	DIMENSION THK(3), TDLX(5), TFLMHL(10), NGETINT 1245		
C	NAMELIST /TITL/ TITLE, NGETINT 1246		
C	NAMELIST /CHANLS/ NSLICE, NSTA, INEDIT, IPILOT, IWRITE, NGETINT 1247	MD1, MD2, MD3, IUNITS, IFILM, IADJIN, NGETINT 1248	
C	NAMELIST /BC/ NBCS, NBCP, BCXS, BCXP, BCHGS, BCHGP, NGETINT 1249	BCTGS, BCTGP, BCQGS, BCQGP, BCPGS, BCPGP, NGETINT 1250	
Z	THUBIN, QHUBIN, TTIPIN, QTIPIN, RHOVG, PEX, BCTIME, TTIO, PTIO, WPLEN, NGETINT 1251	THUBIN, QHUBIN, TTIPIN, QTIPIN, RHOVG, PEX, BCTIME, TTIO, PTIO, WPLEN, NGETINT 1252	
Z	NGETINT 1253	NGETINT 1254	
Z	NGETINT 1255	NGETINT 1255	
Z	NGETINT 1256	NGETINT 1256	
Z	NGETINT 1257	NGETINT 1257	
Z	NGETINT 1258	NGETINT 1258	
Z	NGETINT 1259	NGETINT 1259	
Z	NGETINT 1260	NGETINT 1260	

Z	WSVST,	AKCTBL,	AKWTBL,	RHOC,	RHOM,	NGETINT	1261					
Z	SPHTC,	SPHTM,	DLYME,	TEPS,	TYMMAX	NGETINT	1262					
C	NAMELIST /CONTRL/ NFWD,					ICOR,	NGEO	NGETINT	1263			
C	NAMELIST /PROPS/ CD,					SPAN,	ADUMP,	DHYD,	APLEN,	RO,	NGETINT	1264
Z		RI,	CIMP1,	CIMP2,	CIMP3,	CIMP4,	CIMP5,			NGETINT	1265	
Z		CIMP6,	CIMP7,	DIMP1,	DIMP2,	DIMP3,	DIMP4,			NGETINT	1266	
Z		DIMP5,	DIMP6							NGETINT	1267	
C	NAMELIST /GEO/ ISTA,					ISTB,	THK,	TDLX,	TDHJ,	TXN,	NGETINT	1268
Z		TDHF,	TRR,	IHC	TDP,	TSP,	TPLMHL			NGETINT	1269	
C	DATA TIKLE// //										NGETINT	1270
C	NSLICE =	THE NO. OF SLICES OF THE BLADE THAT ARE BEING CONSIDERED								NGETINT	1271	
C	IHUB =	1	INDICATES A SPECIFIED TEMPERATURE DISTRIBUTION IS GIVEN AT THE HUB END (F)							NGETINT	1272	
C		= 2	INDICATES AN ADIABATIC SURFACE AT THE HUB END							NGETINT	1273	
C		= 3	INDICATES HEAT FLUX IS SPECIFIED AT HUB END (BTU/HR FT**2 R)							NGETINT	1274	
C	ITIP =	1	INDICATES A SPECIFIED TEMPERATURE DISTRIBUTION IS GIVEN AT THE TIP END (F)							NGETINT	1275	
C		= 2	INDICATES AN ADIABATIC SURFACE AT THE TIP END							NGETINT	1276	
C		= 3	INDICATES HEAT FLUX IS SPECIFIED AT TIP END (BTU/HR FT**2 R)							NGETINT	1277	
C	IADJIN =	0,	MEANS TO HOLD PTIO CONSTANT AND ADJUST WPLEN;							NGETINT	1278	
C	> 0,	MEANS TO FIX WPLEN AND ADJUST PTIO.							NGETINT	1279		
C	ISTA =	FIRST STATION NUMBER FOR THIS DATA SET							NGETINT	1280		
C	IF ISTB IS SPECIFIED, IT IS THE LAST STATION NUMBER FOR THIS DATA SET								NGETINT	1281		
C	IF ISTB IS SPECIFIED, IT MUST BE EQUAL TO ISTA + A MULTIPLE OF 2.								NGETINT	1282		
C	THK =	(1)-COATING THICKNESS, (2)-METAL THICKNESS, AND (3)-CHANNEL							NGETINT	1283		
C	WIDTH. ALL IN INCHES.								NGETINT	1284		
C	TDLX =	DISTANCE FROM UPSTREAM NODE (INCHES)							NGETINT	1285		
C	TDHJ =	HYDRAULIC DIAMETER OF IMPINGEMENT JET HOLE (INCHES) - STORED UNDER STATION NUMBER							NGETINT	1286		
C	TDHF =	EFFECTIVE DIAMETER OF FILM COOLING HOLE IF PRESENT (INCHES) - STORED UNDER STATION NUMBER							NGETINT	1287		
C		= DIAMETER OF ONE HOLE*SQRT(No. OF HOLES AT THIS STATION IN THIS SLICE)							NGETINT	1288		
C	TXN =	SPANWISE SPACING OF IMPINGEMENT JETS (INCHES)							NGETINT	1289		
C	TRR =	RADIAL LOCATION OF THIS STATION (INCHES)							NGETINT	1290		
C	IHC	INDICATES THE TYPE OF INSIDE HEAT TRANSFER AT THIS STATION,							NGETINT	1291		
C	= 1	FOR IMPINGEMENT WITH CROSSFLOW							NGETINT	1292		
C	= 2	FOR FORCED CONVECTION CHANNEL FLOW							NGETINT	1293		
C	= 3	FOR PIN FINS							NGETINT	1294		
C	TDP =	THE PIN FIN DIAMETER (IN) IF PINS ARE USED;							NGETINT	1295		
C	TSF =	THE PIN FIN SPACING (IN), ASSUMING AN EQUILATERAL TRIANGULAR ARRAY OF PINS.							NGETINT	1296		
C	AKCTBL=	TABLE OF CLADDING THERMAL CONDUCTIVITY (BTU/HR FT R) VS TEMPERATURE (F)							NGETINT	1297		
C	AKWTBL=	TABLE OF WALL METAL THERMAL CONDUCTIVITY (BTU/HR FT R) VS TEMPERATURE (F)							NGETINT	1298		
C	RHOVG =	HOT GAS FREE STREAM MASS VELOCITY, DENSITY*VELOCITY, FOR FILM COOLING USE, AT EACH FILM COOLING STATION.							NGETINT	1299		
C		INPUT IN (LBM/SEC FT**2), OR (KG/SEC M**2) IF IUNITS=1							NGETINT	1300		
C	RHOC =	DENSITY OF OUTER COATING (LBM/FT**3)							NGETINT	1301		
C	RHOM =	DENSITY OF WALL METAL (LBM/FT**3)							NGETINT	1302		
C	SPHTC =	SPECIFIC HEAT OF COATING (BTU/LBM R)							NGETINT	1303		
C	SPHTM =	SPECIFIC HEAT OF WALL METAL							NGETINT	1304		

```

C DLTIME = TIME STEP SIZE FOR TRANSIENT CALCULATIONS (SEC) NGETINT 1321
C TYMMAX = MAX. TIME (SEC) TO WHICH TRANSIENT IS CARRIED. NGETINT 1322
C TEPS = FRACTION OF TIME STEP AT WHICH TEMP. IS EVALUATED. ( NEW = OLDNGETINT 1323
C           + TEPS*(NEW-OLD)) NGETINT 1324
C WSVST = TABLE OF WHEEL SPEED VS TIME, (RPM VS SEC), ODD SUBSCRIPTS NGETINT 1325
C           ARE SPEED, EVEN ARE TIME, WSVST(2)=0.0 NGETINT 1326
C NGETINT 1327
C CIMP1 TO CIMP5 ARE EXPONENTS TO BE USED IN A GENERAL NGETINT 1328
C           IMPINGEMENT WITH CROSSFLOW CORRELATION. IF NOT SPECIFIED, THEN THENGETINT 1329
C           BUILT IN KIRCHER-TABAKOFF CORRELATION IS USED. NGETINT 1330
C SEE SUBROUTINE HCOOLT FOR DESCRIPTION OF GENEAL CORRELATION. NGETINT 1331
C NGETINT 1332
C INITIALIZE: NGETINT 1333
C NGETINT 1334
100  CONTINUE NGETINT 1335
    IEND = 0 NGETINT 1336
    IADJIN = 0 NGETINT 1337
    IHUB = 2 NGETINT 1338
    ITIP = 2 NGETINT 1339
    CIMP1 = 0.0 NGETINT 1340
    DIMP1 = 0.0 NGETINT 1341
    ADUMP = 0.0 NGETINT 1342
    IFILM = 0 NGETINT 1343
    IUNITS = 2 NGETINT 1344
    ALPHA = .04 NGETINT 1345
    BETA = -.16 NGETINT 1346
    DELTA = 16. NGETINT 1347
    EPS = -1. NGETINT 1348
    CD = .8 NGETINT 1349
C NGETINT 1350
C GAS CONSTANT FOR AIR, FT LBF/LBM R NGETINT 1351
    R = 53.35 NGETINT 1352
C NGETINT 1353
C-- SET VALUES FOR UNITS CORRECTION FACTORS--- NGETINT 1354
C-- ... (1) CONVERTS FROM SI TO ENGLISH, ... (2) MAKES NO CONVERSION-- NGETINT 1355
C           ALREADY IN ENGLISH NGETINT 1356
C NGETINT 1357
C--- CINCH(1) IS CONVERSION FACTOR FROM (CM) TO (IN) NGETINT 1358
    CINCH(1) = .39370 NGETINT 1359
    CINCH(2) = 1.0 NGETINT 1360
C--- CHTC(1) IS CONVERSION FACTOR FROM (WATTS/M**2 K) TO (BTU/HR FT**2R) NGETINT 1361
    CHTC(1) = .17623 NGETINT 1362
    CHTC(2) = 1.0 NGETINT 1363
C--- CHFLX(1) IS CONVERSION FACTOR FROM (WATTS/M**2) TO (BTU/HR FT**2) NGETINT 1364
    CHFLX(1) = .31721 NGETINT 1365
    CHFLX(2) = 1.0 NGETINT 1366
C--- CPRSP(1) IS CONVERSION FACTOR FROM (KILOPASCALS) TO (PSIA) NGETINT 1367
    CPRSR(1) = .14503 NGETINT 1368
    CPRSF(2) = 1.0 NGETINT 1369
C--- CMSFL(1) IS CONVERSION FACTOR FROM (KG/HR) TO (LBM/HR) NGETINT 1370
    CMSFL(1) = 2.67924 NGETINT 1371
    CMSFL(2) = 1.0 NGETINT 1372
C--- CTMPP(1) IS CONVERSION FACTOR FROM (K) TO (E) NGETINT 1373
    CTMPP(1) = 1.8 NGETINT 1374
    CTMPP(2) = 1.0 NGETINT 1375
C--- CTCON(1) IS CONVERSION FACTOR FROM (WATTS/M K) TO (BTU/HR FT R) NGETINT 1376
    CTCON(1) = .57817 NGETINT 1377
    CTCON(2) = 1.0 NGETINT 1378
C--- CDEN(1) IS CONVERSION FACTOR FROM (KG/M**3) TO (LBM/FT**3) NGETINT 1379
    CDEN(1) = .06243 NGETINT 1380

```

	CDEN(2) = 1.0	NGETINT 1381
C---	CSPHT(1) IS CONVERSION FACTOR FROM (J/KG K) TO (BTU/LBM R)	NGETINT 1382
	CSPHT(1) = .000239	NGETINT 1383
	CSPHT(2) = 1.0	NGETINT 1384
C---	CVISC(1) IS CONVERSION FACTOR FROM (PA SEC) TO (LBM/FT HR)	NGETINT 1385
	CVISC(1) = 2419.096	NGETINT 1386
	CVISC(2) = 1.0	NGETINT 1387
C---	CGASC(1) IS CONVERSION FROM (J/KG K) TO (FT LBF/LBM R)	NGETINT 1388
	CGASC(1) = .18602	NGETINT 1389
	CGASC(2) = 1.0	NGETINT 1390
C---	CRHOVG IS CONVERSION FROM (KG/SEC M**2) TO (LBM/SEC FT**2)	NGETINT 1391
	CRHOVG(1) = .0204823	NGETINT 1392
	CRHOVG(2) = 1.0	NGETINT 1393
C	DO 105 I = 1,30	NGETINT 1394
105	TITLE(I) = TIKLE	NGETINT 1395
C	DO 106 I = 1,1000	NGETINT 1396
	BCHGS(I) = 0.0	NGETINT 1397
	BCHGP(I) = 0.0	NGETINT 1398
	BCTGS(I) = 0.0	NGETINT 1399
	BCTGP(I) = 0.0	NGETINT 1400
	BCQGS(I) = 0.0	NGETINT 1401
	BCQGP(I) = 0.0	NGETINT 1402
	BCPGS(I) = 0.0	NGETINT 1403
106	BCPGP(I) = 0.0	NGETINT 1404
	RHOC = 0.0	NGETINT 1405
	RHOM = 0.0	NGETINT 1406
	SPHTC = 0.0	NGETINT 1407
	SPHTM = 0.0	NGETINT 1408
	DO 107 I = 1,400	NGETINT 1409
	THUBIN(I) = 0.0	NGETINT 1410
	QHUBIN(I) = 0.0	NGETINT 1411
	TTIPIN(I) = 0.0	NGETINT 1412
	QTIPIN(I) = 0.0	NGETINT 1413
	RHOVG(I) = 0.0	NGETINT 1414
107	PEX(I) = 0.0	NGETINT 1415
	DO 108 I = 1,50	NGETINT 1416
	BCTIME(I) = 0.0	NGETINT 1417
	TTIO(I) = 0.0	NGETINT 1418
	PTIO(I) = 0.0	NGETINT 1419
	WSVST(I) = 0.0	NGETINT 1420
108	RR(I) = 0.0	NGETINT 1421
C	DO 110 I = 1,6000	NGETINT 1422
110	CHANL(I) = 0.0	NGETINT 1423
C	DO 112 I = 1,15	NGETINT 1424
	PEXIT(I) = 0.0	NGETINT 1425
	DO 112 J = 1,80	NGETINT 1426
	AKC(I,J) = 0.0	NGETINT 1427
112	AKW(I,J) = 0.0	NGETINT 1428
C	DO 115 I = 1,2000	NGETINT 1429
115	INDCHN(I) = 0	NGETINT 1430
	DO 116 I = 1,20	NGETINT 1431
	AKCTBL(I) = 0.0	NGETINT 1432
116	AKWTBL(I) = 0.0	NGETINT 1433
	IPLOT = 0	NGETINT 1434
	IWRITE = 0	NGETINT 1435
		NGETINT 1436
		NGETINT 1437
		NGETINT 1438
		NGETINT 1439
		NGETINT 1440

```

INEDIT = 0 NGETINT 1441
TEPS = 1.0 NGETINT 1442
DLTYME = 0.0 NGETINT 1443
READ(5,TITL)
READ(5,CHANLS)
READ(5,BC)
IF (BCHGS(1).EQ.0.0) BTA=1.0 NGETINT 1444
IP (BCQGS(1).EQ.0.0) BTA = 0.0 NGETINT 1445
IP (TTIPIN(1).GT.0.0) ITIP = 1 NGETINT 1446
IF (THUBIN(1).GT.0.0) IHUB = 1 NGETINT 1447
IF (ABS(QTIPIN(1)).GT.0.0) ITIP = 3 NGETINT 1448
IF (ABS(QHUBIN(1)).GT.0.0) IHUB = 3 NGETINT 1449
WS = WSVST(1) NGETINT 1450
C NGETINT 1451
PEXIT(1) = PEX(1) NGETINT 1452
DO 175 ICHLNO = 1,NSLICE NGETINT 1453
C NGETINT 1454
ICHLNO IS THE CHANNEL NUMBER; = 1 AT THE HUB, = NSLICE AT THE TIP NGETINT 1455
C NGETINT 1456
C READ(5,CTRL) NGETINT 1457
NODSF = 5*NFWD NGETINT 1458
C--NODSF IS THE NUMBER OF NODES IN THE FORWARD REGION NGETINT 1459
C NGETINT 1460
NODST = 5*NSTA NGETINT 1461
C--NODST IS THE TOTAL NUMBER OF NODES IN THE BLADE SLICE ICHLNO NGETINT 1462
C NGETINT 1463
NBLKSZ = (15 + 2*NODST) + 8*NSTA NGETINT 1464
C--NBLKSZ IS THE SIZE OF THE DATA BLOCK RESERVED IN CHANL ARRAY FOR THIS NGETINT 1465
C SLICE ICHLNO NGETINT 1466
C NGETINT 1467
ISBLOK = IEND + 1 NGETINT 1468
C--ISBLOK IS THE STARTING POINT IN CHANL ARRAY FOR THIS BLOCK OF DATA NGETINT 1469
C NGETINT 1470
INSTRT = 15 + (ICHLNO-1)*(15 + NSTA) NGETINT 1471
C--INSTRT IS THE STARTING POINT IN INDCHN ARRAY FOR THIS BLOCK OF NGETINT 1472
C INTEGER DATA NGETINT 1473
C NGETINT 1474
INDCHN(ICHLNO) = INSTRT NGETINT 1475
INDCHN(INSTRT) = ICHLNO NGETINT 1476
INDCHN(INSTRT+1) = IFILM NGETINT 1477
INDCHN(INSTRT+2) = ICOR NGETINT 1478
INDCHN(INSTRT+3) = NFWD NGETINT 1479
INDCHN(INSTRT+4) = NSTA NGETINT 1480
INDCHN(INSTRT+5) = ISBLOK NGETINT 1481
INDCHN(INSTRT+6) = NBLKSZ NGETINT 1482
INDCHN(INSTRT+7) = IPLOT NGETINT 1483
INDCHN(INSTRT+8) = MD1 NGETINT 1484
INDCHN(INSTRT+9) = MD2 NGETINT 1485
INDCHN(INSTRT+10) = MD3 NGETINT 1486
INDCHN(INSTRT+12) = IHUB NGETINT 1487
INDCHN(INSTRT+13) = ITIP NGETINT 1488
IIHCTZ = INSTRT + 14 NGETINT 1489
C--IIHCTZ IS THE RELATIVE ZERO POINT IN INDCHN FOR STORAGE OF THE NGETINT 1490
C INDICATOR IHC NGETINT 1491
READ(5,PROPS)
S(ICHLNO) = SPAN*CINCH(IUNITS) NGETINT 1492
APLN(ICHLNO) = APLEN*CINCH(IUNITS)*CINCH(IUNITS) NGETINT 1493
DPLN(ICHLNO) = DHYD*CINCH(IUNITS) NGETINT 1494
ROUT(ICHLNO) = RO*CINCH(IUNITS) NGETINT 1495
RIN(ICHLNO) = RI*CINCH(IUNITS) NGETINT 1496

```

```

C      NGETINT 1501
C      NOW, /GEO/ IS READ, NGE0 TIMES, AND THE DATA STORED IN CHANL ARRAY.
C      NGETINT 1502
C      NGETINT 1503
C      NGETINT 1504
C      ISBLOK = THE STARTING POINT IN CHANL ARRAY FOR CHANNEL ICHLNO DATA
C      NGETINT 1505
C      FIRST, STORE THE SINGLE VALUED DATA
C      NGETINT 1506
C      NGETINT 1507
C      NGETINT 1508
C      CHANL(ISBLOK) = CD
C      NGETINT 1509
C      CHANL(ISBLOK+1) = ALPHA
C      NGETINT 1510
C      CHANL(ISBLOK+2) = BETA
C      NGETINT 1511
C      CHANL(ISBLOK+3) = DELTA
C      NGETINT 1512
C      CHANL(ISBLOK+4) = EPS
C      NGETINT 1513
C      CHANL(ISBLOK+6) = ADUMP*CINCH(IUNITS)**2
C      NGETINT 1514
C      CHANL(ISBLOK+7) = SPAN*CINCH(IUNITS)
C      NGETINT 1515
C      CHANL(ISBLOK+8) = BTA
C      NGETINT 1516
C      CHANL(ISBLOK+9) = DLTYME
C      NGETINT 1517
C      CHANL(ISBLOK+10)= TEPS
C      NGETINT 1518
C      NGETINT 1519
C      THEN THE ARRAYS ARE STORED:
C      NGETINT 1520
C      NGETINT 1521
C      THE FOLLOWING ARE STORED BY NODE NUMBER:
C      NGETINT 1522
C      THK (TAU), TDLX (DLX)
C      NGETINT 1523
C      THE REST ARE STORED BY STATION NUMBER:
C      NGETINT 1524
C      TDHJ (DHJ), TDHF (DHF), TXN (XN), TRR (RR).
C      NGETINT 1525
C      TDP (DP), TSP (SP),
C      NGETINT 1526
C      (AKC), (AKW), IHCT (IHC).
C      NGETINT 1527
C      NGETINT 1528
C      ITHKZ = ISBLOK + 14
C      NGETINT 1529
C      ITDLXZ = ISBLOK + 14 + NODST
C      NGETINT 1530
C      ITDHJZ = ISBLOK + 14 + 2*NODST
C      NGETINT 1531
C      ITDHFZ = ISBLOK + 14 + 2*NODST + NSTA
C      NGETINT 1532
C      ITXNZ = ISBLOK + 14 + 2*NODST + 2*NSTA
C      NGETINT 1533
C      ITERZ = ISBLOK + 14 + 2*NODST + 3*NSTA
C      NGETINT 1534
C      ITDPZ = ISBLOK + 14 + 2*NODST + 4*NSTA
C      NGETINT 1535
C      ITSPZ = ISBLOK + 14 + 2*NODST + 5*NSTA
C      NGETINT 1536
C      IEEND = ISBLOK + 14 + 2*NODST + 8*NSTA
C      NGETINT 1537
C      THK(1) = 0.0
C      NGETINT 1538
C      DO 170 I = 1,NGEO
C      NGETINT 1539
C      ISTB = 0
C      NGETINT 1540
C      READ(5,GEO)
C      NGETINT 1541
C      IF (THK(1).LE.0.0) THK(1) = .0001*THK(2)
C      NGETINT 1542
C      IF (TDLX(1).GT.2.0*TDLX(4).OR.TDLX(4).GT.1.2*TDLX(1))
C      NGETINT 1543
C      Z           WRITE(8,136) ICHLNO,ISTA
C      NGETINT 1544
C      136 FORMAT(/' CHANNEL ',I2,', STATION ',I3,
C      NGETINT 1545
C      ' , ---TDLX VALUES DO NOT LOOK RIGHT')
C      NGETINT 1546
C      Z           IF (TDHJ.GT.0..AND.TXN.LT.1.1*TDHJ) WRITE(8,137) ICHLNO,ISTA
C      NGETINT 1547
C      137 FORMAT(/' CHANNEL ',I2,', STATION ',I3,
C      NGETINT 1548
C      ' , ---HOLE SPACING AND DIAMETER DO NOT LOOK RIGHT')
C      NGETINT 1549
C      IP (ISTB.EQ.0) ISTB = ISTA
C      NGETINT 1550
C      DO 165 J = ISTA,ISTB,2
C      NGETINT 1551
C      NGETINT 1552
C      C      J REPRESENTS THE STATION NUMBER IN THIS CASE
C      NGETINT 1553
C      C      JSENS = J - 2*(J/2)
C      NGETINT 1554
C      C      JSENS = 0 INDICATES THAT STATION NO. IS EVEN AND STATION IS ON
C      NGETINT 1555
C      SUCTION SIDE
C      NGETINT 1556
C      C      JSENS = 1 INDICATES THAT STATION NO. IS ODD AND STATION IS ON
C      NGETINT 1557
C      PRESSURE SIDE
C      NGETINT 1558
C      NGETINT 1559
C      NGETINT 1560

```

```

C
IARG = ITDHJZ + J          NGETINT 1561
CHANL(IARG) = TDHJ*CINCH(IUNITS)  NGETINT 1562
IARG = ITDHFZ + J          NGETINT 1563
CHANL(IARG) = TDHF*CINCH(IUNITS)  NGETINT 1564
IARG = ITXNZ + J          NGETINT 1565
CHANL(IARG) = TXN*CINCH(IUNITS)  NGETINT 1566
IARG = ITRRZ + J          NGETINT 1567
CHANL(IARG) = TRR*CINCH(IUNITS)  NGETINT 1568
IARG = ITDPZ + J          NGETINT 1569
CHANL(IARG) = TDP*CINCH(IUNITS)  NGETINT 1570
IARG = ITSPZ + J          NGETINT 1571
CHANL(IARG) = TSP*CINCH(IUNITS)  NGETINT 1572
IARG = IIHCTZ + J          NGETINT 1573
INDCHN(IARG) = IHCT        NGETINT 1574
NODOUT = 5*J - 4          NGETINT 1575
NGETINT 1576
NGETINT 1577
NGETINT 1578
NGETINT 1579
NGETINT 1580
NGETINT 1581
NGETINT 1582
NGETINT 1583
NGETINT 1584
NGETINT 1585
NGETINT 1586
NGETINT 1587
NGETINT 1588
NGETINT 1589
NGETINT 1590
NGETINT 1591
NGETINT 1592
NGETINT 1593
NGETINT 1594
NGETINT 1595
NGETINT 1596
NGETINT 1597
NGETINT 1598
NGETINT 1599
NGETINT 1600
NGETINT 1601
NGETINT 1602
NGETINT 1603
NGETINT 1604
NGETINT 1605
NGETINT 1606
NGETINT 1607
NGETINT 1608
NGETINT 1609
NGETINT 1610
NGETINT 1611
NGETINT 1612
NGETINT 1613
NGETINT 1614
NGETINT 1615
NGETINT 1616
NGETINT 1617
NGETINT 1618
NGETINT 1619
NGETINT 1620

C      NODOUT IS THE NODE NO. ON THE OUTSIDE SURFACE AT STATION J
C      5 IS THE NUMBER OF NODES AT STATION J
C
145    CONTINUE
      LOCA = ITDLXZ + NODOUT
      IF (TDLX(3).LE.0.) GO TO 155
      DO 150 L = 1,5
      LOCAL = LOCA + L - 1
150    CHANL(LOCAL) = TDLX(L)*CINCH(IUNITS)
      GO TO 160
155    CHANL(LOCA) = TDLX(1)*CINCH(IUNITS)
      CHANL(LOCA+3) = TDLX(4)*CINCH(IUNITS)
      AA = TDLX(1)
      B = (TDLX(4)-TDLX(1))/(THK(1)+THK(2))
      CHANL(LOCA+1) = (AA + B*THK(1))*CINCH(IUNITS)
      CHANL(LOCA+2) = (AA + B*(THK(1)+THK(2)/2.))*CINCH(IUNITS)
      CHANL(LOCA+4) = (AA + B*(THK(1)+THK(2)+THK(3)/2.))*CINCH(IUNITS)
160    CONTINUE
      LOCA = ITHKZ + NODOUT
      CHANL(LOCA) = THK(1)*CINCH(IUNITS)
      CHANL(LOCA+2) = THK(2)*CINCH(IUNITS)
      CHANL(LOCA+4) = THK(3)*CINCH(IUNITS)
165    CONTINUE
170    CONTINUE
175    CONTINUE
C
C---  CONVERT UNITS ON BC DATA
C
      IF (IUNITS.EQ.2) GO TO 300
      NTBC = 1
      DO 205 I = 2,50
      IF (BCTIME(I).LE.0.0) GO TO 210
205    NTBC = NTBC + 1
210    NSETS = NBCS*NSLICE*NTBC
      NSETP = NBCP*NSLICE*NTBC
C
      DO 215 I = 1,NSETS
      BCXS(I) = BCXS(I)*CINCH(1)
      BCHGS(I) = BCHGS(I)*CHTC(1)
      BCTGGS(I) = BCTGGS(I)*CTMPP(1) - 460.
      BCQGGS(I) = BCQGGS(I)*CHFLX(1)
      BCPGGS(I) = BCPGGS(I)*CPRSR(1)
215

```

```

DO 220 I = 1,NSETP          NGETINT 1621
BCXP(I) = BCXP(I)*CINCH(1)  NGETINT 1622
BCHCP(I) = BCHGP(I)*CHTC(1) NGETINT 1623
BCTGP(I) = BCTGP(I)*CTMPF(1) - 460. NGETINT 1624
BCQGP(I) = BCQGP(I)*CHFLX(1)  NGETINT 1625
220   BCPGP(I) = BCPGP(I)*CPRS(I)  NGETINT 1626
C
NSET = NSTA*NTBC           NGETINT 1627
DO 225 I = 1,NSET          NGETINT 1628
RHOVG(I) = RHOVG(I)*CRHOVG(1) NGETINT 1629
THUBIN(I) = THUBIN(I)*CTMPF(1) - 460. NGETINT 1630
QHUBIN(I) = QHUBIN(I)*CHFLX(1)  NGETINT 1631
TTIPIN(I) = TTIPIN(I)*CTMPF(1) - 460. NGETINT 1632
225   QTIPIN(I) = QTIPIN(I)*CHFLX(1) NGETINT 1633
C
NSET = NSLICE*NTBC          NGETINT 1634
DO 230 I = 1,NSET          NGETINT 1635
230   PEX(I) = PEX(I)*CPRS(IUNITS) NGETINT 1636
C
DO 235 I = 1,49,2          NGETINT 1637
TTIO(I) = TTIO(I)*CTMPF(IUNITS) - 460. NGETINT 1638
235   PTIO(I) = PTIO(I)*CPRS(IUNITS) NGETINT 1639
WPLEN = WPLEN*CMSFL(IUNITS)    NGETINT 1640
RHOC = RHOC*CDEN(IUNITS)      NGETINT 1641
RHOM = RHOM*CDEN(IUNITS)      NGETINT 1642
SPHTC = SPHTC*CSPHT(IUNITS)   NGETINT 1643
SPHTM = SPHTM*CSPHT(IUNITS)   NGETINT 1644
C
DO 280 I = 1,19,2          NGETINT 1645
AKCTBL(I) = AKCTBL(I)*CTMPF(IUNITS) - 460. NGETINT 1646
AKCTBL(I+1) = AKCTBL(I+1)*CTCON(IUNITS) NGETINT 1647
AKWTBL(I) = AKWTBL(I)*CTMPF(IUNITS) - 460. NGETINT 1648
280   AKWTBL(I+1) = AKWTBL(I+1)*CTCON(IUNITS) NGETINT 1649
C
300   CONTINUE               NGETINT 1650
C
IF (IFILM.NE.2) GO TO 320   NGETINT 1651
DO 310 I = 1,NSTA          NGETINT 1652
310   RHOVGA(I) = RHOVG(I)    NGETINT 1653
C
320   CONTINUE               NGETINT 1654
C
C IF INEDIT .GT. 0, PRINT AN INPUT EDIT
C
DO 180 I = 1,NSLICE         NGETINT 1655
180   CALL INPRT(I,INEDIT)    NGETINT 1656
185   CONTINUE               NGETINT 1657
RETURN
END
NGETINT 1658
NGETINT 1659
NGETINT 1660
NGETINT 1661
NGETINT 1662
NGETINT 1663
NGETINT 1664
NGETINT 1665
NGETINT 1666
NGETINT 1667
NGETINT 1668
NGETINT 1669
NGETINT 1670

```

```

C----SOURCE.NHCFRCT
FUNCTION HCFRCT(S,LCOOL,LIN)
C
C- SOURCE.NHCFRCT-----
COMMON /TCO/ ADUMP,      BTA,      CD,      CP,
Z        GAM,       PIM,      R,       SPAN,      TOG,
Z        WDUMP,      WIM,      AKC(15,80), AKW(15,80),
Z        A(400),     AJET(80), AM2(80),   CNUM(80),
Z        DH(80),     DHF(80),  DHJ(80),
NHCFRCT 1671
NHCFRCT 1672
NHCFRCT 1673
NHCFRCT 1674
NHCFRCT 1675
NHCFRCT 1676
NHCFFRCT 1677
NHCFRCT 1678
NHCFRCT 1679
NHCFRCT 1680

```

```

Z      DLX(400),  FF(80),      HC(80),      HG(80),      NHCFRCT 1681
Z      P(2,15,80), PEXIT(15),  PUMP(80),    OG(80),      NHCFRCT 1682
Z      QSNK(80),   RR(80),      S(15),       T(2,15,400), NHCFRCT 1683
Z      TG(80),     TAU(400),    WFC(80),    XN(80),      NHCFRCT 1684
Z      WJ(15,80),  WCROS(2,15,80),  IHUB,       ITIP,      NHCFRCT 1685
Z      ICOR,       IFILM,      NBLKSZ,     NSLICE,     NHCFRCT 1686
Z      ISBLOK,     ISLICE,      NSTA,       IHC(80)    NHCFRCT 1687
Z      NFWD,        NSTA,      NHCFRCT 1688
Z      NHCFRCT 1689
C      COMPUTE TURBULENT HEAT TRANSFER COEFFICIENT IN CHANNEL FLOW:
C      NU = .023*( RE**.8 )*( PD**.333 )
C
100  CONTINUE
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
      FE = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)
      HCFRCD = .023*12.*(C/DH(IS))*(RE**.8)*(PD**.333)
200  CONTINUE
      RETURN
      END

C----SOURCE.NHCOOLT
      SUBROUTINE NHCOOL(JS)
C
C- SOURCE.NHCOOLT---
C
      COMMON /IMPCOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7,
Z          DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6
C
      COMMON /PRPS/ CPO,      GAMO,      DP(80),      SP(80),      RE(80),
Z          CPC(80),    GAMC(80),    DUMR1(80),    DUMR2(80)
C
      COMMON /TCO/ ADUMP,     BTA,       CD,        CP,
Z          GAM,       FIM,       R,        SPAN,      TOG,
Z          WDUMP,     WIM,       AKC(15,80),  AKW(15,80), NHCOOLT 1701
Z          A(400),    AJET(80),   AM2(80),    CNUM(80),  NHCOOLT 1702
Z          DH(80),    DHF(80),   DHJ(80),    NHCOOLT 1703
Z          DLX(400),  FF(80),   HC(80),    HG(80),    NHCOOLT 1704
Z          P(2,15,80), PEXIT(15), PUMP(80), OG(80),    NHCOOLT 1705
Z          QSNK(80),  RR(80),   S(15),     T(2,15,400), NHCOOLT 1706
Z          TG(80),    TAU(400), WFC(80),    XN(80),    NHCOOLT 1707
Z          WJ(15,80),  WCROS(2,15,80), IHUB,     ITIP,    NHCOOLT 1708
Z          ICOR,     IFILM,    NBLKSZ,   NSLICE,   NHCOOLT 1709
Z          ISBLOK,   ISLICE,   NSTA,     IHC(80)  NHCOOLT 1710
Z          NFWD,     NSTA,    NHCOOLT 1711
Z          NHCOOLT 1712
C
      DIMENSION IGG(80), IRE(80), REJ(80), REJOVR(80)
1      CONTINUE
      TMP=TOG
      CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
      CONDCT = C
      XMUTOG = XMU
      PDTOG = PD
      PI=3.14159
      IF (JS.GT.1) GO TO 101
      IF (ICOR.EQ.1) GO TO 101
      IF (WJ(ISLICE,JS).LE.0.0) GO TO 101
C
C--- LEADING EDGE HEAT TRANSFER CORRELATION FOR STATIONS FORWARD OF ICORNHCOOLT 1739
C--- CORRELATION OF METZGER ET AL, J. ENG. POWER, JULY 1969, PP 149-158 NHCOOLT 1740

```

```

C
5      NEND=ICUR+1
      XS = 0.0
      XP = 0.0
      DO 50 J = 3,NEND,2
      LINS = 5*(J-1) - 1
      LINP = 5*J - 1
      XS = XS + DLX(LINS)
      XP = XP + DLX(LINP)
50    CONTINUE
55    XL = (XS + XP)/2.
      IF (AJET(JS).GT.0.) GMASS = WJ(ISLICE,JS)/AJET(JS)
60    BES = PI*DHJ(JS)**2/(4.*XN(JS))
      DEH = 2.*BES
65    REJ(JS) = 12.*3600.*GMASS*DEH/XMU
      PROD = REJ(JS)**.27*(XL/BES)**.52
70    STANMX = .355/PROD
      HC(JS) = STANMX*CP*GMASS*144.*3600.
C
      IF (REJ(JS).LT.1150..OR.REJ(JS).GT.6300.) WRITE(6,75) REJ(JS)
75    FORMAT(1H /'***WARNING*** LEADING EDGE IMPINGEMENT JET REYNOLDS ',NHC00LT 1741
      Z      'NUMBER IS ',F8.1/'      RANGE OF THE CORRELATION IS 1150',-,NHC00LT 1742
      Z      '< REJ < 6300')
      ILEAD = ICOR - 1
      IF (ILEAD.LT.2) GO TO 85
      DO 80 I = 2,ILEAD
      IF(WJ(ISLICE,I).GT.0.0) GO TO 90
80    HC(I) = HC(JS)
85    CONTINUE
     GO TO 101
90    WRITE(8,95) ICOR
95    FORMAT(//'      SOLUTION TERMINATED***TOO MANY ROWS OF IMPINGEMENT',
      Z      '      HOLES FORWARD OF STATION',I3,'. HOLES ARE ',
      Z      'ALLOWED ONLY AT STATION 1.')
      STOP
C
C--KIRCHER-TABAKOFF CORRELATION, IMPINGEMENT WITH CROSS FLOW
C--ICOR = STATION NUMBER APPLICATION OF THIS CORRELATION BEGINS
C
101   IGGC = 0
      IREC = 0
      ISTRT=ICOR
      IF (JS.GT.1) ISTRT= 1
C
      IF (CIMP1.NE.0.0) GO TO 400
      DO 130 I = ISTRT,NFWD
      WC = ABS(WCRROS(2,ISLICE,I))
      II = 5*I
      REJ(I) = 0.0
      IF (IHC(I).EQ.1) GO TO 103
      LCOOL = 5*I
      LIN = LCOOL - 1
      RC(I) = HCFRCD(I,LCOOL,LIN)
      GO TO 130
103   CONTINUE
      IF (AJET(I).EQ.0.0) GO TO 128
      IF (WJ(ISLICE,I).LE.0.) GO TO 128
      TMP=(T(2,ISLICE,LIN)+T0G)/2.
      CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
      CONDCT = C
      NHC00LT 1743
      NHC00LT 1744
      NHC00LT 1745
      NHC00LT 1746
      NHC00LT 1747
      NHC00LT 1748
      NHC00LT 1749
      NHC00LT 1750
      NHC00LT 1751
      NHC00LT 1752
      NHC00LT 1753
      NHC00LT 1754
      NHC00LT 1755
      NHC00LT 1756
      NHC00LT 1757
      NHC00LT 1758
      NHC00LT 1759
      NHC00LT 1760
      NHC00LT 1761
      NHC00LT 1762
      NHC00LT 1763
      NHC00LT 1764
      NHC00LT 1765
      NHC00LT 1766
      NHC00LT 1767
      NHC00LT 1768
      NHC00LT 1769
      NHC00LT 1770
      NHC00LT 1771
      NHC00LT 1772
      NHC00LT 1773
      NHC00LT 1774
      NHC00LT 1775
      NHC00LT 1776
      NHC00LT 1777
      NHC00LT 1778
      NHC00LT 1779
      NHC00LT 1780
      NHC00LT 1781
      NHC00LT 1782
      NHC00LT 1783
      NHC00LT 1784
      NHC00LT 1785
      NHC00LT 1786
      NHC00LT 1787
      NHC00LT 1788
      NHC00LT 1789
      NHC00LT 1790
      NHC00LT 1791
      NHC00LT 1792
      NHC00LT 1793
      NHC00LT 1794
      NHC00LT 1795
      NHC00LT 1796
      NHC00LT 1797
      NHC00LT 1798
      NHC00LT 1799
      NHC00LT 1800

```

```

XHUTOG = XMU          NHCOOLT 1801
PDTOG = PD            NHCOOLT 1802
105 REJ(I)=WJ(ISLICE,I)/AJET(I)*DHJ(I)/(XHUTOG/3600.)*12.0
GG=(WC/A(II))/(WJ(ISLICE,I)/AJET(I))
IF (GG.LE.2.0) GO TO 110
IGGC = IGGC + 1
IGG(IGGC) = I
110 CONTINUE
IF (REJ(I).GE.300.0.AND.REJ(I).LE.3.E4) GO TO 115
IREC = IREC + 1
IRE(IREC) = I
115 CONTINUE
IF (REJ(I).LT.3000.) GO TO 120
AM=-.002517*(XN(I)/DHJ(I))**2+.068485*XN(I)/DHJ(I)+.506994
HC(I)=REJ(I)**AM
HC(I)=HC(I)*EXP(.02596*(XN(I)/DHJ(I))**2-.8259*XN(I)/DHJ(I)+.3985)
HC(I)=HC(I)/(1.+.4696*)
Z   ((WC/A(II))/(WJ(ISLICE,I)/AJET(I))*TAU(II)/DHJ(I))**.965) NHCOOLT 1818
GO TO 125             NHCOOLT 1819
120 AM=-.001452*(XN(I)/DHJ(I))**2+.042838*(XN(I)/DHJ(I))+.516548 NHCOOLT 1820
HC(I)=REJ(I)**AM
HC(I)=HC(I)*EXP(.0126*(XN(I)/DHJ(I))**2-.5106*XN(I)/DHJ(I)-.2057) NHCOOLT 1821
HC(I)=HC(I)/(1.+.4215*)
Z   ((WC/A(II))/(WJ(ISLICE,I)/AJET(I))*TAU(II)/DHJ(I))**.58) NHCOOLT 1822
125 CONTINUE           NHCOOLT 1823
HC(I)=HC(I)*CONDCT/DHJ(I)*12.0*PDTOG**.33*(TAU(II)/DHJ(I))**.091 NHCOOLT 1824
GO TO 130             NHCOOLT 1825
C
128 CONTINUE           NHCOOLT 1826
IF (I.GT.2) HC(I) = HC(I-2) NHCOOLT 1827
IF (I.EQ.2) HC(I) = HC(1) NHCOOLT 1828
IF (I.EQ.1) HC(I) = HC(3) NHCOOLT 1829
130 CONTINUE           NHCOOLT 1830
IST = NFWD + 1         NHCOOLT 1831
DO 150 I = IST,NSTA,2 NHCOOLT 1832
IF (IHC(I).NE.1) GO TO 155 NHCOOLT 1833
150 HC(I) = HC(I-2)     NHCOOLT 1834
155 IST = NFWD+2        NHCOOLT 1835
DO 160 I = IST,NSTA,2 NHCOOLT 1836
IF (IHC(I).NE.1) GO TO 165 NHCOOLT 1837
160 HC(I) = HC(I-2)     NHCOOLT 1838
165 CONTINUE           NHCOOLT 1839
IF (IGGC.GT.0) WRITE(6,140) (IGG(I),I=1,IGGC) NHCOOLT 1840
DO 132 I = 1,IREC      NHCOOLT 1841
ISTATN = IRE(I)         NHCOOLT 1842
REJOVR(I) = REJ(ISTATN) NHCOOLT 1843
132 CONTINUE           NHCOOLT 1844
IF (IREC.GT.0) WRITE(6,145) (IRE(I),REJOVR(I),I=1,IREC) NHCOOLT 1845
135 CONTINUE           NHCOOLT 1846
140 FORMAT(1H /* ***** WARNING ***** RATIO OF CROSSFLOW TO *, NHCOOLT 1847
Z   'JET-FLOW IS OUT OF THE RANGE OF ', NHCOOLT 1848
Z   'THE CORRELATION AT THE FOLLOWING STATIONS: '/23X,20(I4,'.')) NHCOOLT 1849
145 FORMAT(1H /* ***** WARNING ***** JET REYNOLD'S NUMBER IS *, NHCOOLT 1850
Z   'OUT OF THE RANGE OF THE CORRELATION ', NHCOOLT 1851
Z   'AT THE FOLLOWING STATIONS: '/1X,8('**',I2,'--',F8.1,'*')) NHCOOLT 1852
DO 301 I = 1,NFWD       NHCOOLT 1853
DUMR2(I) = REJ(I)       NHCOOLT 1854
301 CONTINUE           NHCOOLT 1855
RETURN                 NHCOOLT 1856
NHCOOLT 1857
NHCOOLT 1858
NHCOOLT 1859
NHCOOLT 1860

```

```

C--- GENERAL CORRELATION FOR IMPINGEMENT WITH CROSSFLOW IS EVALUATED HERHCOOLT 1861
C   E
C--- FORM OF CORRELATION IS:
C--- ST = CIMP1*(GG**CIMP2)*(GI**CIMP3)*((Z/D)**CIMP4)
C       *((X/D)**CIMP5)*(REJ**CIMP6)*(PDTOG**CIMP7)
C--- WHERE GG IS THE MASS FLUX RATIO, FREE STREAM TO JET, AND
C--- GI IS THE MOMENTUM FLUX RATIO.
C
400  CONTINUE
    DO 450 I = ISTRT,NFWD
        WC = ABS(WCROS(2,ISLICE,I))
        II = 5*I
        ROINVJ = R*T0G/(144.*P(2,ISLICE,I))
        ROINVC = R*T(2,ISLICE,II)/(144.*P(2,ISLICE,I))
        REJ(I) = 0.0
        IF (IHC(I).EQ.1) GO TO 403
        LCOOL = 5*I
        LIN = LCOOL - 1
        HC(I) = HCFRCRD(I,LCOOL,LIN)
        GO TO 450
403  CONTINUE
        IF (AJET(I).EQ.0.0) GO TO 445
        IF (WJ(ISLICE,I).LE.0.) GO TO 445
        TMP=(T(2,ISLICE,LIN)+T0G)/2.
        CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
        CONDCT = C
        XMUTOG = XMU
        PDTOG = PD
405  REJ(I)=WJ(ISLICE,I)/AJET(I)*DHJ(I)/(XMUTOG/3600.)*12.0
        GG=(WC/A(II))/(WJ(ISLICE,I)/AJET(I))
        GI = ((WC/A(II))**2*ROINVC)/((WJ(ISLICE,I)/AJET(I))**2*ROINVJ)
        ZOVERD = TAU(II)/DHJ(I)
        XOVERD = XN(I)/DHJ(I)
        ST = CIMP1*(GG**CIMP2)*(GI**CIMP3)*(ZOVERD**CIMP4)
        Z     *(XOVERD**CIMP5)*(REJ(I)**CIMP6)*(PDTOG**CIMP7)
        HC(I) = 144.*3600.*ST*CP*WJ(ISLICE,I)/AJET(I)
        GO TO 450
C
445  CONTINUE
        IF (I.GT.2) HC(I) = HC(I-2)
        IF (I.EQ.2) HC(I) = HC(1)
        IF (I.EQ.1) HC(I) = HC(3)
450  CONTINUE
        IST = NFWD+1
        DO 460 I = IST,NSTA,2
        IF (IHC(I).NE.1) GO TO 465
460  HC(I) = HC(I-2)
465  IST = NFWD+2
        DO 470 I = IST,NSTA,2
        IF (IHC(I).NE.1) GO TO 475
470  HC(I) = HC(I-2)
475  CONTINUE
C
        DO 485 I = 1,NFWD
        DUMR2(I) = REJ(I)
485  CONTINUE
        RETURN
        END

```

C----SOURCE.NHCPINT
SUBROUTINE HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EPAREA)

NHCPINT 1919
NHCPINT 1920

```

C- SOURCE.NHCPINT---- NHCPI NT 1921
C- COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80), NHCPI NT 1922
Z CPC(80), GAMC(80), DUMR1(80), DUMR2(80) NHCPI NT 1923
C- COMMON /TCO/ ADUMP, BTA, CD, CP, NHCPI NT 1924
Z GAM, PIM, R, SPAN, TOG, NHCPI NT 1925
Z WDUMP, WIM, AKC(15,80), AKW(15,80), NHCPI NT 1926
Z A(400), AJET(80), AM2(80), CNUM(80), NHCPI NT 1927
Z DH(80), DHF(80), DHJ(80), NHCPI NT 1928
Z DLX(400), FP(80), HC(80), HG(80), NHCPI NT 1929
Z P(2,15,80),PEXIT(15), PUMP(80), QG(80), NHCPI NT 1930
Z QSNK(80), RR(80), S(15), T(2,15,400), NHCPI NT 1931
Z TG(80), TAU(400), WFC(80), XN(80), NHCPI NT 1932
Z WJ(15,80), WCROS(2,15,80), NHCPI NT 1933
Z ICOR, IFILM, IHUB, ITIP, NHCPI NT 1934
Z ISBLOK, ISLICE, NBLKSZ, NSLICE, NHCPI NT 1935
Z NFWD, NSTA, IHC(80) NHCPI NT 1936
C- DIMENSION EPAREA(80), DELTAN(15) NHCPI NT 1937
C- COMPUTE THE HEAT TRANSFER COEFFICIENT AND EFFECTIVENESS FOR A NHCPI NT 1938
C- TRIANGULAR ARRAY OF PIN PINS NHCPI NT 1939
C- WHERE DP IS PIN DIAMETER IN INCHES AND SP IS PIN SPACING IN INCHES NHCPI NT 1940
C- VDP = DP(IS) NHCPI NT 1941
C- VSP = SP(IS) NHCPI NT 1942
100 CONTINUE NHCPI NT 1943
    TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2. NHCPI NT 1944
    CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU) NHCPI NT 1945
C-- NO. OF PINS AT THIS STATION IS: NHCPI NT 1946
    PINS = SPAN*DLX(LCOOL)/(.86603*VSP**2) NHCPI NT 1947
C-- AVERAGE LENGTH OF PINS: NHCPI NT 1948
    SLP = (TAU(LCOOL) + TAU(LCUP))/2. NHCPI NT 1949
    IF (IS.GT.NFWD.AND.IS.LE.NFWD+2) SLP = TAU(LCOOL) NHCPI NT 1950
C-- MINIMUM FLOW AREA: NHCPI NT 1951
    AMIN = SLP*SPAN*(VSP-VDP)/VSP NHCPI NT 1952
C-- TOTAL SURFACE AREA: NHCPI NT 1953
    PHTTR = 2.*DLX(LCOOL)*SPAN + 3.14159*PINS*(VDP*SLP-VDP**2/4.) NHCPI NT 1954
C-- CHANNEL HYDRAULIC DIAMETER: NHCPI NT 1955
    DH(IS) = 4.*AMIN*DLX(LCOOL)/AHTTR NHCPI NT 1956
    REDH = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(AMIN*XNU) NHCPI NT 1957
    TERM1 = -.89*(VSP/SLP)**.5075 NHCPI NT 1958
    TERM2 = -3.094*VDP/VSP NHCPI NT 1959
    TERM3 = 4.143*EXP(TERM1 + TERM2)/(REDH**.2946) NHCPI NT 1960
105 CONTINUE NHCPI NT 1961
    HC(IS) = (12.*C/DH(IS))*(.023 + TERM3)*(REDH**.8)*(PD**.333) NHCPI NT 1962
    EML = SQRT(4.*HC(IS)*SLP**2/(AKW(ISLICE,IS)*VDP)) NHCPI NT 1963
    EPTVNS = TANH(EML)/EML NHCPI NT 1964
C-- CHECK LOCATION OF HEAT FLOW SPLIT POINT IF THIS IS A TRAILING NHCPI NT 1965
C- EDGE REGION STATION NHCPI NT 1966
C- IF (IS.LE.NFWD) GO TO 160 NHCPI NT 1967
    TBAR = (T(2,ISLICE,LCOOLP)-T(2,ISLICE,LCOOL))/ NHCPI NT 1968
Z (T(2,ISLICE,LIN)-T(2,ISLICE,LCOOL)) NHCPI NT 1969
    HYCOS = COSH(EML) NHCPI NT 1970
    HYSIN = SINH(EML) NHCPI NT 1971
    IF (HYCOS-TBAR.LT.HYSIN) GO TO 120 NHCPI NT 1972

```

```

110  WRITE(6,110) LCOOLP,LIN,DELTAN(ISLICE)          NHCPIINT 1981
      FORMAT(1H //' **** WARNING **** NODE',I3,
      Z       ' IS RECEIVING HEAT FROM NODE',I3,' THROUGH THE PINS.',
      Z       ' RESULTS ARE INVALID. DELTAN =',F7.4)        NHCPIINT 1982
      GO TO 140                                         NHCPIINT 1983
120  CONTINUE                                         NHCPIINT 1984
      IF (HYCOS-TBAR.GT.0.) GO TO 130                 NHCPIINT 1985
      WRITE(6,110) LIN,LCOOLP,DELTAN(ISLICE)          NHCPIINT 1986
      GO TO 140                                         NHCPIINT 1987
130  CONTINUE                                         NHCPIINT 1988
      XOVRL = (HYCOS-TEAR)/HYSIN                     NHCPIINT 1989
      XOVRL = ALOG((1.+XOVRL)/(1.-XOVRL))/(2.*EML)   NHCPIINT 1990
140  CONTINUE                                         NHCPIINT 1991
160  CONTINUE                                         NHCPIINT 1992
      EFAREA(IS) = DLX(LIN)*SPAN                      NHCPIINT 1993
      Z       - 3.14159*PINS*(VDP**2/4.- EPTVNS*VDP*SLP*XOVRL) NHCPIINT 1994
      IF (IS.GT.NPWD) EFAREA(IS+1) = DLX(LCOOLP)*SPAN   NHCPIINT 1995
      Z       - 3.14159*PINS*(VDP**2/4.- EPTVNS*VDP*SLP*(1.-XOVRL)) NHCPIINT 1996
      IF (IS.LE.NPWD) EFAREA(IS) = DLX(LIN)*SPAN       NHCPIINT 1997
      Z       - 3.14159*PINS*(VDP**2/4.- EPTVNS*VDP*SLP)  NHCPIINT 1998
170  CONTINUE                                         NHCPIINT 1999
      RETURN                                           NHCPIINT 2000
      END                                              NHCPIINT 2001
                                                NHCPIINT 2002
                                                NHCPIINT 2003

```

```

C-----SOURCE.NINPRTT
      SUBROUTINE INPRT(ICHLN,INEDIT)                  NINPRTT 2004
C
C----- SOURCE.NINPRTT-----
C
      COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(100C), NINPRTT 2005
      Z      BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NINPRTT 2006
      Z      BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),   NINPRTT 2007
      Z      QHUBIN(400), QHUB(80),   TTIPIN(400), TTIP(80),   NINPRTT 2008
      Z      QTIPIN(400), QTIP(80),   RHOVG(400), PEX(400),   NINPRTT 2009
      Z      BCTIME(50),   TTIO(50),   PTIO(50),   WPLEN,     NINPRTT 2010
      Z      WSVST(50),   AKCTBL(20), AKWTBL(20), NBCS, NBCP NINPRTT 2011
C
      COMMON /GAAS/ GS(200), NG                       NINPRTT 2012
C
      COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80), NINPRTT 2013
      Z      CPC(80), GAMC(80), DUMR1(80), DUMR2(80)  NINPRTT 2014
C
      COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15), NINPRTT 2015
      Z      PIN(15), TIN(15), W(15), WS               NINPRTT 2016
      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NINPRTT 2017
      Z      IPLOT, MD1, MD2, MD3, IADJIN, IWRITE    NINPRTT 2018
C
      COMMON /TCO/ ADUMP, BTA, CD, CP,                 NINPRTT 2019
      Z      GAM, PIM, R, SPAN, TOG,                 NINPRTT 2020
      Z      WDUMP, WIM, AKC(15,80), AKW(15,80),   NINPRTT 2021
      Z      A(400), AJET(80), AM2(80), CNUM(80),   NINPRTT 2022
      Z      DH(80), DHF(80), DHJ(80),               NINPRTT 2023
      Z      DLX(400), FF(80), HC(80), HG(80),     NINPRTT 2024
      Z      P(2,15,80), PEXIT(15), PUMP(80), OG(80),   NINPRTT 2025
      Z      QSNK(80), RR(80), S(15), T(2,15,400),   NINPRTT 2026
      Z      TG(80), TAU(400), WFC(80), XN(80),     NINPRTT 2027
      Z      WJ(15,80), WCROS(2,15,80),               NINPRTT 2028
      Z      ICOR, IFILM, IHUB, ITIP,                 NINPRTT 2029
      Z      ISBLOK, ISLICE, NBLKSZ, NSLICE,           NINPRTT 2030
      Z      NPWD, NSTA, IHC(80)                      NINPRTT 2031
C
                                                NINPRTT 2032
                                                NINPRTT 2033
                                                NINPRTT 2034
                                                NINPRTT 2035
                                                NINPRTT 2036
                                                NINPRTT 2037
                                                NINPRTT 2038
                                                NINPRTT 2039
                                                NINPRTT 2040

```

```

COMMON /TRNSNT/ RHOC,          RHOM,          SPHTC,          SPHTB,
Z      DLTIME,        TIME,        TEPS,        TIMEMAX          NINPRTT 2041
C
COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSPL(2),
Z      CTMPP(2), CTCOM(2), CDEN(2), CSPHT(2), CGASC(2),
Z      CVISC(2), CRHOVG(2), IUNITS          NINPRTT 2042
NINPRTT 2043
C
DIMENSION DUM1(10),DUM2(10),DUM3(10),DUM4(10),DUM5(10),DUM6(10),
Z      DUM7(10),DUM8(10),DUM9(10),DUM53(10),DUM55(10)          NINPRTT 2044
DIMENSION DUM10(10),DUM11(10),DUM12(10),DUM13(10),DUM14(10),
Z      DUM15(10),DUM16(10),DUM25(10),DUM52(10)          NINPRTT 2045
DIMENSION DUM17(10),DUM18(10),DUM19(10),DUM20(10)          NINPRTT 2046
NINPRTT 2047
NINPRTT 2048
NINPRTT 2049
NINPRTT 2050
NINPRTT 2051
NINPRTT 2052
NINPRTT 2053
NINPRTT 2054
NINPRTT 2055
NINPRTT 2056
NINPRTT 2057
NINPRTT 2058
NINPRTT 2059
NINPRTT 2060
NINPRTT 2061
NINPRTT 2062
NINPRTT 2063
NINPRTT 2064
NINPRTT 2065
NINPRTT 2066
NINPRTT 2067
NINPRTT 2068
NINPRTT 2069
NINPRTT 2070
NINPRTT 2071
NINPRTT 2072
NINPRTT 2073
NINPRTT 2074
NINPRTT 2075
NINPRTT 2076
NINPRTT 2077
NINPRTT 2078
NINPRTT 2079
NINPRTT 2080
NINPRTT 2081
NINPRTT 2082
NINPRTT 2083
NINPRTT 2084
NINPRTT 2085
NINPRTT 2086
NINPRTT 2087
NINPRTT 2088
NINPRTT 2089
NINPRTT 2090
NINPRTT 2091
NINPRTT 2092
NINPRTT 2093
NINPRTT 2094
NINPRTT 2095
NINPRTT 2096
NINPRTT 2097
NINPRTT 2098
NINPRTT 2099
NINPRTT 2100
C
INITIALIZE TEMPERATURE DISTRIBUTION (DEGREES R)
C
I = ICHNL
NODSF = 5*NFWD
NODSTM = 5*NSTA - 4
NODST = 5*NSTA
DO 830 I1 = 5,NODST,5
IS = I1/5
LO = I1-4
LJ = I1-3
L = I1-2
LI = I1-1
T(2,I,LO) = .9*TG(IS)
T(2,I,LI) = T(2,I,LO)/1.08
T(2,I,LJ) = T(2,I,LO) - (T(2,I,LO)-T(2,I,LI))*TAU(LO)/(TAU(LO) +
Z                                         TAU(L))          NINPRTT 2061
NINPRTT 2062
NINPRTT 2063
NINPRTT 2064
NINPRTT 2065
NINPRTT 2066
NINPRTT 2067
NINPRTT 2068
NINPRTT 2069
NINPRTT 2070
NINPRTT 2071
NINPRTT 2072
NINPRTT 2073
NINPRTT 2074
NINPRTT 2075
NINPRTT 2076
NINPRTT 2077
NINPRTT 2078
NINPRTT 2079
NINPRTT 2080
NINPRTT 2081
NINPRTT 2082
NINPRTT 2083
NINPRTT 2084
NINPRTT 2085
NINPRTT 2086
NINPRTT 2087
NINPRTT 2088
NINPRTT 2089
NINPRTT 2090
NINPRTT 2091
NINPRTT 2092
NINPRTT 2093
NINPRTT 2094
NINPRTT 2095
NINPRTT 2096
NINPRTT 2097
NINPRTT 2098
NINPRTT 2099
NINPRTT 2100
930  T(2,I,I1) = TTIO(1) + 460.
ISTR = NODSF + 5
DO 860 I1 = ISTR,NODSTM,10
T(2,I,I1) = T(2,I,NODSF)
T(2,I,I1+5) = T(2,I,I1)
DO 860 J = 1,4
IPJ = I1 + J
IMJ = I1 + J - 5
IUPP = NODSF + J - 5
T(2,I,IPJ) = T(2,I,IUPP)
T(2,I,IMJ) = T(2,I,IUPP)
860  DO 865 J = 1,NODST
865  T(1,I,J) = T(2,I,J)
C
IF (ICHNL.GT.1) GO TO 94
WRITE(6,408)
408  FORMAT(1H1,///,20X,'PROPERTY TABLES'///)
WRITE(6,410)
410  FORMAT(1H , 'OUTER COATING EFFECTIVE THERMAL CONDUCTIVITY')
C
IF (IUNITS.EQ.1) GO TO 420
C
WRITE(6,412) (AKCTBL(I),I=1,19,2)
WRITE(6,414) (AKCTBL(I),I=2,20,2)

```

```

412  FORMAT(//5X,'T, (F)',10X,10(F9.1))          NINPRTT 2101
414  FORMAT(5X,'K, (BTU/HR/FT/R)',10(F9.3))      NINPRTT 2102
416  WRITE(6,416)                                     NINPRTT 2103
416  FORMAT(////' WALL METAL THERMAL CONDUCTIVITY') NINPRTT 2104
416  WRITE(6,412) (AKWTBL(I),I=1,19,2)            NINPRTT 2105
416  WRITE(6,414) (AKWTBL(I),I=2,20,2)            NINPRTT 2106
C
C     GO TO 445
C
420  CONTINUE
DO 418 I = 1,19,2
AKCTBL(I) = (AKCTBL(I)+460.)/1.8
AKWTBL(I) = (AKWTBL(I)+460.)/1.8
AKCTBL(I+1) = AKCTBL(I+1)/CTCON(1)
418  AKWTBL(I+1) = AKWTBL(I+1)/CTCON(1)
C
WRITE(6,422) (AKCTBL(I),I=1,19,2)          NINPRTT 2110
WRITE(6,424) (AKCTBL(I),I=2,20,2)          NINPRTT 2111
422  FORMAT(//5X,'T, (K)',4X,10(F9.1))        NINPRTT 2112
424  FORMAT(5X,'K, (W/M/K)',10(F9.3))        NINPRTT 2113
WRITE(6,416)                                     NINPRTT 2114
WRITE(6,422) (AKWTBL(I),I=1,19,2)          NINPRTT 2115
WRITE(6,424) (AKWTBL(I),I=2,20,2)          NINPRTT 2116
C
C
DO 448 I = 1,19,2
AKCTBL(I) = 1.8*AKCTBL(I) - 460.
AKWTBL(I) = 1.8*AKWTBL(I) - 460.
AKCTBL(I+1) = AKCTBL(I+1)*CTCON(1)
448  AKWTBL(I+1) = AKWTBL(I+1)*CTCON(1)
C
445  CONTINUE
C
WRITE(6,450)                                     NINPRTT 2124
450  FORMAT(1H ////' TABLE OF GAS PROPERTIES')   NINPRTT 2125
NGS = NG
IP (NG.GT.10) NGS = 10
C
IP (IUNITS.EQ.1) GO TO 470
C
WRITE(6,452) (GS(J),J=1,NGS)                  NINPRTT 2126
452  FORMAT(//5X,'TEMPERATURE (F)',10(F9.1))   NINPRTT 2127
L = NG + 1
LE = NG + NGS
WRITE(6,454) (GS(J),J=L,LE)                  NINPRTT 2128
454  FORMAT(5X,'K, (BTU/HR/FT/R)',10(F9.5))   NINPRTT 2129
L = 2*NG + 1
LE = 2*NG + NGS
WRITE(6,456) (GS(J),J=L,LE)                  NINPRTT 2130
456  FORMAT(5X,'CP, (BTU/LBM/R)',10(F9.5))   NINPRTT 2131
L = 3*NG + 1
LE = 3*NG + NGS
WRITE(6,458) (GS(J),J=L,LE)                  NINPRTT 2132
458  FORMAT(5X,'PRANDTL NUMBER',10(F9.5))    NINPRTT 2133
L = 4*NG + 1
LE = 4*NG + NGS
WRITE(6,460) (GS(J),J=L,LE)                  NINPRTT 2134
460  FORMAT(5X,'VIS. (LBM/FT/HR)',10(F9.5))  NINPRTT 2135
C
GO TO 90

```

```

C
C
470  CONTINUE
DO 471 J = 1,NGS
471  DUM1(J) = (GS(J)+460.)/1.8
      WRITE(6,472) (DUM1(J),J=1,NGS)
472  FORMAT(//5X,'TEMPERATURE (K)',10(F9.1))
      L = NG + 1
      LE = NG + NGS
      JI = 0
      DO 473 J = L,LE
      JI = JI+1
473  DUM1(JI) = GS(J)/CTCON(1)
      WRITE(6,474) (DUM1(J),J=1,JI)
474  FORMAT(5X,'K,          (W/M/K)',10(F9.5))
      L = 2*NG + 1
      LE = 2*NG + NGS
      JI = 0
      DO 475 J = L,LE
      JI = JI+1
475  DUM1(JI) = GS(J)/CSPHT(1)
      WRITE(6,476) (DUM1(J),J=1,JI)
476  FORMAT(5X,'CP,          (J/KG/K)',10(F9.2))
      L = 3*NG + 1
      LE = 3*NG + NGS
      WRITE(6,478) (GS(J),J=L,LE)
478  FORMAT(5X,'PRANDTL NUMBER',10(F9.5))
      L = 4*NG + 1
      LE = 4*NG + NGS
      JI = 0
      DO 479 J = L,LE
      JI = JI+1
479  DUM1(JI) = GS(J)/CVISC(1)
      WRITE(6,480) (DUM1(J),J=1,JI)
480  FORMAT(5X,'VIS. (N S/M**2)',10(F9.5))
C
90   CONTINUE
      IF (INEDIT.EQ.0) GO TO 350
C
C-- LIST OUT THE INPUT H/T GAS BOUNDARY CONDITIONS--
C
C--MNBC IS THE MAX OF (NBCS & NBCP)
C
      MNBC = NBCS
      IF (MNBC.LT.NBCP) MNBC=NBCP
      NTIMES = 1
C--NTIMES IS THE NUMBER OF TIME STEPS IN BC TABLES
481  IF (BCTIME(NTIMES+1).LE.0.0) GO TO 482
      NTIMES = NTIMES + 1
      GO TO 481
C
482  CONTINUE
      WRITE(6,4820)
4820 FORMAT(1H1,40X,'HOT GAS BOUNDARY CONDITIONS')
      WRITE(6,483)
483  FORMAT(' *****SUCTION SIDE*****',22X,
              '*****PRESSURE SIDE*****')
C
C--SET THE NO. OF POINTS PER TIME STEP IN S&P BC ARRAYS
      NPRTS = NSLICE*NBCS
      NINPRTT 2161
      NINPRTT 2162
      NINPRTT 2163
      NINPRTT 2164
      NINPRTT 2165
      NINPRTT 2166
      NINPRTT 2167
      NINPRTT 2168
      NINPRTT 2169
      NINPRTT 2170
      NINPRTT 2171
      NINPRTT 2172
      NINPRTT 2173
      NINPRTT 2174
      NINPRTT 2175
      NINPRTT 2176
      NINPRTT 2177
      NINPRTT 2178
      NINPRTT 2179
      NINPRTT 2180
      NINPRTT 2181
      NINPRTT 2182
      NINPRTT 2183
      NINPRTT 2184
      NINPRTT 2185
      NINPRTT 2186
      NINPRTT 2187
      NINPRTT 2188
      NINPRTT 2189
      NINPRTT 2190
      NINPRTT 2191
      NINPRTT 2192
      NIEPRTT 2193
      NINPRTT 2194
      NINPRTT 2195
      NINPRTT 2196
      NINPRTT 2197
      NINPRTT 2198
      NINPRTT 2199
      NINPRTT 2200
      NINPRTT 2201
      NINPRTT 2202
      NINPRTT 2203
      NINPRTT 2204
      NINPRTT 2205
      NINPRTT 2206
      NINPRTT 2207
      NINPRTT 2208
      NINPRTT 2209
      NINPRTT 2210
      NINPRTT 2211
      NINPRTT 2212
      NINPRTT 2213
      NINPRTT 2214
      NINPRTT 2215
      NINPRTT 2216
      NINPRTT 2217
      NINPRTT 2218
      NINPRTT 2219
      NINPRTT 2220

```

```

NPRTP = NSLICE*NBCP          NINPRTT 2221
C
NL = 3                         NINPRTT 2222
DO 499 IT = 1,NTIMES          NINPRTT 2223
C
C-- SET THE NO. OF POINTS THAT PRECEDED TIME STEP 'IT'
NPRCS = NPRTS*(IT-1)          NINPRTT 2224
NPRCP = PRTP*(IT-1)           NINPRTT 2225
C
C--START THE LOOP THROUGH ALL SLICES      NINPRTT 2226
DO 499 ISL = 1,NSLICE         NINPRTT 2227
C
C--SET THE NO. OFF POINTS PRECEDING THIS SLICE   NINPRTT 2228
NBFRS = NBFRS + NBCS*(ISL-1)  NINPRTT 2229
NBFRP = NBFRP + NBCP*(ISL-1)  NINPRTT 2230
C
NL = NL + 3 + MNBC           NINPRTT 2231
IF (NL.LT.60) GO TO 4860      NINPRTT 2232
NL = 3 + MNBC                 NINPRTT 2233
WRITE(6,4820)                  NINPRTT 2234
WRITE(6,483)                   NINPRTT 2235
4860 CONTINUE                  NINPRTT 2236
C
IF (IT.EQ.1) WRITE(6,484) ISL  NINPRTT 2237
484 FORMAT(45X,'INITIAL STEADY STATE'/49X,'SLICE NO.',I2)  NINPRTT 2238
IF (IT.GT.1) WRITE(6,485) BCTIME(IT), ISL                NINPRTT 2239
485 FORMAT(45X,'BCTIME =',F8.3,' SEC'/47X,'SLICE NO.',I2)  NINPRTT 2240
WRITE(6,486)                  NINPRTT 2241
486 FORMAT(4X,'X',7X,'HG',6X,'TG',10X,'QG',6X,'PG',
Z      26X,'X',7X,'HG',6X,'TG',10X,'QG',6X,'PG')  NINPRTT 2242
C
C--HERE WE LOOP WITHIN A SLICE          NINPRTT 2243
C
DO 499 IBC = 1,MNBC           NINPRTT 2244
IF (IBC.GT.NBCS) GO TO 487     NINPRTT 2245
J = NBFRS + IBC               NINPRTT 2246
JXS = (ISL-1)*NBCS + IBC      NINPRTT 2247
TBCXS = BCXS(JXS)/CINCH(IUNITS)  NINPRTT 2248
TBCHGS = BCHGS(J)/CHTC(IUNITS)  NINPRTT 2249
TBCTGS = BCTGS(J)             NINPRTT 2250
IF (IUNITS.EQ.1) TBCTGS=(BCTGS(J)+460.)/1.8  NINPRTT 2251
TBCQGS = BCQGS(J)/CHFLX(IUNITS)  NINPRTT 2252
TBCPGS = BCPGS(J)/CPRSR(IUNITS)  NINPRTT 2253
WRITE(6,489) TBCXS,TBCHGS,TBCTGS,TBCQGS,TBCPGS  NINPRTT 2254
489 FORMAT(2X,F6.2,2F8.1,F12.1,F8.1)  NINPRTT 2255
487 CONTINUE                    NINPRTT 2256
IF (IBC.GT.NBCP) GO TO 499     NINPRTT 2257
J = NBFRP + IBC               NINPRTT 2258
JXP = (ISL-1)*NBCP + IBC      NINPRTT 2259
TBCXP = BCXP(JXP)/CINCH(IUNITS)  NINPRTT 2260
TBCHGP = BCHGP(J)/CHTC(IUNITS)  NINPRTT 2261
TBCTGP = BCTGP(J)             NINPRTT 2262
IF (IUNITS.EQ.1) TBCTGP=(BCTGP(J)+460.)/1.8  NINPRTT 2263
TBCQGP = BCQGP(J)/CHFLX(IUNITS)  NINPRTT 2264
TBCPGP = BCPGP(J)/CPRSR(IUNITS)  NINPRTT 2265
IF (IBC.LE.NBCS) WRITE(6,488) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP  NINPRTT 2266
IF (IBC.GT.NBCS) WRITE(6,490) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP  NINPRTT 2267
488 FORMAT(1H+,65X,F6.2,2F8.1,F12.1,F8.1)  NINPRTT 2268
490 FORMAT(66X,F6.2,2F8.1,F12.1,F8.1)  NINPRTT 2269
499 CONTINUE                    NINPRTT 2270

```

```

94    CONTINUE          NINPRTT 2281
DO 95 I = 1,200      NINPRTT 2282
95    NFLUID(I) = 0    NINPRTT 2283
100   WRITE(6,150) ICHNL NINPRTT 2284
     IF (ICHNL.GT.1) GO TO 101 NINPRTT 2285
     IF (IHUB.EQ.1) WRITE(6,142) NINPRTT 2286
     IF (IHUB.EQ.2) WRITE(6,144) NINPRTT 2287
     IF (IHUB.EQ.3) WRITE(6,146) NINPRTT 2288
101   CONTINUE          NINPRTT 2289
     IF (ICHNL.LT.NSLICE) GO TO 102 NINPRTT 2290
     IF (ITIP.EQ.1) WRITE(6,147) NINPRTT 2291
     IF (ITIP.EQ.2) WRITE(6,148) NINPRTT 2292
     IF (ITIP.EQ.3) WRITE(6,149) NINPRTT 2293
102   CONTINUE          NINPRTT 2294
     TRIN = RIN(ICHLN)/CINCH(IUNITS) NINPRTT 2295
     TROUT = ROUT(ICHLN)/CINCH(IUNITS) NINPRTT 2296
     TDPLN = DPLN(ICHLN)/CINCH(IUNITS) NINPRTT 2297
     TAPLN = APLN(ICHLN)/(CINCH(IUNITS)*CINCH(IUNITS)) NINPRTT 2298
     WRITE(6,103) TRIN, UL(IUNITS), TROUT, UL(IUNITS), NINPRTT 2299
Z         TDPLN, UL(IUNITS), TAPLN, UA(IUNITS) NINPRTT 2300
103   FORMAT(/' COOLANT PLENUM: RI=',F7.3,A4,' RO=',F7.3,A4,4X, NINPRTT 2301
Z           'DHYD=',F7.4,A4,' APLEN=',F7.4,A4,'*2') NINPRTT 2302
C
C       IF (IUNITS.EQ.1) GO TO 500 NINPRTT 2303
C
      WRITE(6,153) NFWD,NSTA,SPAN NINPRTT 2304
      WRITE(6,155) CD,ADUMP NINPRTT 2305
      TEM = TTIO(1) + 460. NINPRTT 2306
      WRITE(6,157) TEM,PTIO(1),PEY(ICHLN),WPLEN NINPRTT 2307
      ITRBG = NFWD + 2 NINPRTT 2308
      WRITE(6,154) ICHNL,ITRBG NINPRTT 2309
      DO 118 I = 1,NSTA,20 NINPRTT 2310
      IP18 = I + 18 NINPRTT 2311
      IF (IP18.GT.NSTA) IP18 = NSTA NINPRTT 2312
      IF (I.EQ.1) WRITE(6,156) (J,J=I,IP18,2) NINPRTT 2313
      IF (I.GT.1) WRITE(6,159) (J,J=I,IP18,2) NINPRTT 2314
      ID = 0 NINPRTT 2315
      DO 104 J = I,IP18,2 NINPRTT 2316
      ID = ID + 1 NINPRTT 2317
      DUM1(ID) = RR(J) NINPRTT 2318
      NFLUID(J) = 5*j NINPRTT 2319
104   CONTINUE          NINPRTT 2320
      WRITE(6,158) (NFLUID(J),J=I,IP18,2) NINPRTT 2321
      ID = 0 NINPRTT 2322
      DO 116 J = I,IP18,2 NINPRTT 2323
      ID = ID + 1 NINPRTT 2324
      NOS = NFLUID(J) - 4 NINPRTT 2325
      IF (NOS.GT.1) GO TO 106 NINPRTT 2326
      XOS = 0.0 NINPRTT 2327
      XJN = 0.0 NINPRTT 2328
      XMN = 0.0 NINPRTT 2329
      XIS = 0.0 NINPRTT 2330
      XCC = 0.0 NINPRTT 2331
      GO TO 108 NINPRTT 2332
106   XOS = XOS + DLX(NOS) NINPRTT 2333
      XJN = XJN + DLX(NOS+1) NINPRTT 2334
      XMN = XMN + DLX(NOS+2) NINPRTT 2335
      XIS = XIS + DLX(NOS+3) NINPRTT 2336
      XCC = XCC + DLX(NOS+4) NINPRTT 2337
108   CONTINUE          NINPRTT 2338
                                NINPRTT 2339
                                NINPRTT 2340

```

DUM2 (ID) = XOS	NINPRTT 2341
DUM25 (ID) = XJM	NINPFTT 2342
DUM3 (ID) = XMM	NINPFTT 2343
DUM4 (ID) = XIS	NINPRTT 2344
DUM5 (ID) = XCC	NINPRTT 2345
DUM55 (ID) = TAU(NOS)	NINPRTT 2346
DUM6 (ID) = TAU(NOS+2)	NINPRTT 2347
NOS = NFLUID(J)	NINPRTT 2348
DUM7 (ID) = TAU(NOS)	NINPRTT 2349
DUM8 (ID) = A(NOS)	NINPRTT 2350
DUM9 (ID) = DH(J)	NINPRTT 2351
DUM10 (ID) = DHJ(J)	NINPRTT 2352
DUM11 (ID) = CNUM(J)	NINPRTT 2353
DUM12 (ID) = AJET(J)	NINPRTT 2354
DUM16 (ID) = THUBIN(J) - 460.	NINPRTT 2355
DUM17 (ID) = QHUBIN(J)	NINPRTT 2356
DUM18 (ID) = TTIPIN(J) - 460.	NINPRTT 2357
DUM19 (ID) = QTIPIN(J)	NINPRTT 2358
NOS = NFLUID(J) - 4	NINPRTT 2359
DUM13 (ID) = TG(J) - 460.	NINPRTT 2360
DUM14 (ID) = HG(J)	NINPRTT 2361
JHCAL = IHC(J)	NINPPTT 2362
DUM15 (ID) = HCAL(JHCAL)	NINPPTT 2363
IF (BTA.GT..01) DUM14 (ID) = QG(J)	NINPPTT 2364
116 CONTINUE	NINPRTT 2365
WRITE(6,160) (DUM1(J),J=1,ID)	NINPRTT 2366
WRITE(6,162) (DUM2(J),J=1,ID)	NINPPTT 2367
WRITE(6,163) (DUM25(J),J=1,ID)	NINPRTT 2368
WRITE(6,164) (DUM3(J),J=1,ID)	NINPPTT 2369
WRITE(6,166) (DUM4(J),J=1,ID)	NINPPTT 2370
WRITE(6,168) (DUM5(J),J=1,ID)	NINPPTT 2371
WRITE(6,169) (DUM55(J),J=1,ID)	NINPRTT 2372
WRITE(6,170) (DUM6(J),J=1,ID)	NINPRTT 2373
WRITE(6,172) (DUM7(J),J=1,ID)	NINPRTT 2374
WRITE(6,174) (DUM8(J),J=1,ID)	NINPRTT 2375
WRITE(6,176) (DUM9(J),J=1,ID)	NINPRTT 2376
WRITE(6,178) (DUM10(J),J=1,ID)	NINPRTT 2377
WRITE(6,180) (DUM11(J),J=1,ID)	NINPPTT 2378
WRITE(6,182) (DUM12(J),J=1,ID)	NINPRTT 2379
WRITE(6,183) (DUM15(J),J=1,ID)	NINPRTT 2380
WRITE(6,184) (DUM13(J),J=1,ID)	NINPRTT 2381
IF (BTA.LT..01) WRITE(6,186) (DUM14(J),J=1,ID)	NINPPTT 2382
IF (BTA.GT..01) WRITE(6,188) (DUM14(J),J=1,ID)	NINPPTT 2383
IF (ICHNL.GT.1) GO TO 118	NINPRTT 2384
IF (IHUB.EQ.1) WRITE(6,196) (DUM16(J),J=1,ID)	NINPPTT 2385
IF (IHUB.EQ.3) WRITE(6,198) (DUM17(J),J=1,ID)	NINPRTT 2386
IF (ITIP.EQ.1) WRITE(6,202) (DUM18(J),J=1,ID)	NINPRTT 2387
IF (ITIP.EQ.3) WRITE(6,204) (DUM19(J),J=1,ID)	NINPPTT 2388
116 CONTINUE	NINPPTT 2389
ITRBG = NFWD + 1	NINPPTT 2390
WRITE(6,190) ICHNL,ITRBG	NINPRTT 2391
XOS = 0.0	NINPRTT 2392
XJM = 0.0	NINPRTT 2393
XMM = 0.0	NINPRTT 2394
XIS = 0.0	NINPRTT 2395
XCC = 0.0	NINPRTT 2396
DO 140 I = 2,NSTA,20	NINPRTT 2397
IP18 = I + 18	NINPRTT 2398
IF (IP18.GT.NSTA) IP18 = NSTA-1	NINPRTT 2399
IF (I.EQ.2) WRITE(6,156) (J,J=I,IP18,2)	NINPRTT 2400

```

IF (I.GT.2) WRITE(6,159) (J,J=J,IP18,2)
ID = 0
DO 122 J = I,IP18,2
ID = ID + 1
DUM1(ID) = RR(J)
NFLUID(J) = 5*J
122 CONTINUE
WRITE(6,158) (NFLUID(J),J=I,IP18,2)
ID = 0
DO 130 J = I,IP18,2
ID = ID + 1
NOS = NFLUID(J) - 4
XOS = XOS + DLX(NOS)
XJN = XJN + DLX(NOS+1)
XMM = XMM + DLX(NOS+2)
XIS = XIS + DLX(NOS+3)
XCC = XCC + DLX(NOS+4)
DUM2(ID) = XOS
DUM25(ID) = XJN
DUM3(ID) = XMM
DUM4(ID) = XIS
DUM5(ID) = XCC
DUM55(ID) = TAU(NOS)
DUM6(ID) = TAU(NOS+2)
NOS = NFLUID(J)
DUM7(ID) = TAU(NOS)
DUM8(ID) = A(NOS)
DUM9(ID) = DH(J)
DUM10(ID) = DHJ(J)
DUM11(ID) = CNUM(J)
DUM12(ID) = AJET(J)
DUM16(ID) = THUBIN(J) - 460.
DUM17(ID) = QHUBIN(J)
DUM18(ID) = TTIPIN(J) - 460.
DUM19(ID) = QTIPIN(J)
NOS = NFLUID(J) - 4
DUM13(ID) = TG(J) - 460.
DUM14(ID) = HG(J)
JHCAL = IHC(J)
DUM15(ID) = HCAL(JHCAL)
IF (BTA.GT..01) DUM14(ID) = QG(J)
130 CONTINUE
WRITE(6,160) (DUM1(J),J=1,ID)
WRITE(6,162) (DUM2(J),J=1,ID)
WRITE(6,163) (DUM25(J),J=1,ID)
WRITE(6,164) (DUM3(J),J=1,ID)
WRITE(6,166) (DUM4(J),J=1,ID)
WRITE(6,168) (DUM5(J),J=1,ID)
WRITE(6,169) (DUM55(J),J=1,ID)
WRITE(6,170) (DUM6(J),J=1,ID)
WRITE(6,172) (DUM7(J),J=1,ID)
WRITE(6,174) (DUM8(J),J=1,ID)
WRITE(6,176) (DUM9(J),J=1,ID)
WRITE(6,178) (DUM10(J),J=1,ID)
WRITE(6,180) (DUM11(J),J=1,ID)
WRITE(6,182) (DUM12(J),J=1,ID)
WRITE(6,183) (DUM15(J),J=1,ID)
WRITE(6,184) (DUM13(J),J=1,ID)
IF (BTA.LT..01) WRITE(6,186) (DUM14(J),J=1,ID)
IF (BTA.GT..01) WRITE(6,188) (DUM14(J),J=1,ID)

NINPRTT 2401
NINPRTT 2402
NINPRTT 2403
NINPRTT 2404
NINPRTT 2405
NINPRTT 2406
NINPRTT 2407
NINPRTT 2408
NINPRTT 2409
NINPRTT 2410
NINPRTT 2411
NINPRTT 2412
NINPRTT 2413
NINPRTT 2414
NINPRTT 2415
NINPRTT 2416
NINPRTT 2417
NINPRTT 2418
NINPRTT 2419
NINPRTT 2420
NINPRTT 2421
NINPRTT 2422
NINPRTT 2423
NINPRTT 2424
NINPRTT 2425
NINPRTT 2426
NINPRTT 2427
NINPRTT 2428
NINPRTT 2429
NINPRTT 2430
NINPRTT 2431
NINPRTT 2432
NINPRTT 2433
NINPRTT 2434
NINPRTT 2435
NINPRTT 2436
NINPRTT 2437
NINPRTT 2438
NINPRTT 2439
NINPRTT 2440
NINPRTT 2441
NINPRTT 2442
NINPRTT 2443
NINPRTT 2444
NINPRTT 2445
NINPRTT 2446
NINPRTT 2447
NINPRTT 2448
NINPRTT 2449
NINPRTT 2450
NINPRTT 2451
NINPRTT 2452
NINPRTT 2453
NINPRTT 2454
NINPRTT 2455
NINPRTT 2456
NINPRTT 2457
NINPRTT 2458
NINPRTT 2459
NINPRTT 2460

```

```

IF (ICHNL.GT.1) GO TO 140           NINPRTT 2461
IF (IHUB.EQ.1) WRITE(6,196) (DUM16(J),J=1,1D)   NINPRTT 2462
IF (IHUB.EQ.3) WRITE(6,198) (DUM17(J),J=1,1D)   NINPRTT 2463
IF (ITIP.EQ.1) WRITE(6,202) (DUM18(J),J=1,1D)   NINPRTT 2464
IF (ITIP.EQ.3) WRITE(6,204) (DUM19(J),J=1,1D)   NINPRTT 2465
140 CONTINUE                         NINPRTT 2466
150 FORMAT(1H1,46X,'INPUT FOR SLICE NUMBER',I3)    NINPRTT 2467
142 FORMAT(21X,' HUB TEMPERATURES ARE SPECIFIED') NINPRTT 2468
144 FORMAT(21X,' ADIABATIC HUB SPECIFIED')        NINPRTT 2469
146 FORMAT(21X,' HUB HEAT FLUX IS SPECIFIED')     NINPRTT 2470
147 FORMAT(21X,' TIP TEMPERATURES ARE SPECIFIED')  NINPRTT 2471
148 FORMAT(21X,' ADIABATIC TIP SPECIFIED')         NINPRTT 2472
149 FORMAT(21X,' TIP HEAT FLUX IS SPECIFIED')     NINPRTT 2473
153 FORMAT(/' NUMBER OF STATIONS IN IMPINGEMENT REGION IS',I3,      NINPRTT 2474
          ' , TOTAL NUMBER OF STATIONS IS',I3,      NINPRTT 2475
          ' , SPAN OF THIS SLICE IS',F6.3,' IN')   NINPRTT 2476
155 FORMAT(' IMPINGEMENT HOLE DISCHARGE COEF.=',F6.3,      NINPRTT 2477
          ' , AREA OF DUMP TO TRAILING EDGE =',F8.5,' IN**2') NINPRTT 2478
157 FORMAT(' COOLANT INLET TEMP.=',F7.1,' R, COOLANT INLET PRESSURE',NINPRTT 2479
          ' =',F6.1,' PSIA, EXIT PRESSURE =',      NINPRTT 2480
          ' F6.1,' PSIA,'// COOLANT FLOW =',F6.1,' LBM/HR') NINPRTT 2481
154 FORMAT(/' PRESSURE SIDE, SLICE ',I2,', TRAILING EDGE REGION ',      NINPRTT 2482
          ' BEGINS AT STATION-',I3)                 NINPRTT 2483
156 FORMAT(/' STATION NUMBER',5X,10(6X,I4))       NINPRTT 2484
158 FORMAT(' COOLANT NODE NUMBER',10(6X,I4))      NINPRTT 2485
159 FORMAT(1H2//' STATION NUMBER',5X,10(6X,I4))   NINPRTT 2486
160 FORMAT(' RADIAL LOCATION(IN)',10F10.3)        NINPRTT 2487
162 FORMAT(' X, OUTSIDE SUR.(IN)',10F10.5)        NINPRTT 2488
163 FORMAT(' X, INTERFACE (IN)',10F10.5)          NINPRTT 2489
164 FORMAT(' X, MID-METAL (IN)',10F10.5)          NINPRTT 2490
166 FORMAT(' X, INSIDE SURF.(IN)',10F10.5)        NINPRTT 2491
168 FORMAT(' X, MID.COOL.CH.(IN)',10F10.5)        NINPRTT 2492
169 FORMAT(/' COATING THKNSS (IN)',10F10.5)       NINPRTT 2493
170 FORMAT(' WALL THICKNESS (IN)',10F10.5)        NINPRTT 2494
172 FORMAT(' CHANNEL WIDTH (IN)',10F10.5)          NINPRTT 2495
174 FORMAT(' CHANNEL AREA(IN**2)',10F10.5)        NINPRTT 2496
176 FORMAT(' CHANNEL HYD.DIA(IN)',10F10.5)        NINPRTT 2497
178 FORMAT(/' IMP.JET HYD.DIA(IN)',10F10.5)       NINPRTT 2498
180 FORMAT(' NO. OF IMP. JETS ',10F10.2)          NINPRTT 2499
182 FORMAT(' TOT.JET AREA(IN**2)',10F10.5)        NINPRTT 2500
183 FORMAT(/' TYPE OF HC CALC. ',10(6X,A4))       NINPRTT 2501
184 FORMAT(' OUTSIDE BC: TG,(F)',10F10.1)         NINPRTT 2502
186 FORMAT(' HG (BTU/HR/FT**2/R)',10F10.1)       NINPRTT 2503
188 FORMAT(' QG (BTU/HR/FT**2)',10F10.1)         NINPRTT 2504
190 FORMAT(1H1,' SUCTION SIDE, SLICE ',I2,', TRAILING EDGE REGION',      NINPRTT 2505
          ' BEGINS AT STATION-',I3)                 NINPRTT 2506
192 FORMAT(/' CLAD K(BTU/HR/FT/R)',10F10.3)       NINPRTT 2507
194 FORMAT(' METL K(BTU/HR/FT/R)',10F10.3)        NINPRTT 2508
196 FORMAT(' GIVEN HUB TEMP. (F)',10F10.1)        NINPRTT 2509
198 FORMAT(' QHUB (BTU/HR/FT**2)',10F10.1)       NINPRTT 2510
202 FORMAT(' GIVEN TIP TEMP. (F)',10F10.1)        NINPRTT 2511
204 FORMAT(' QTIP (BTU/HR/FT**2)',10F10.1)       NINPRTT 2512
C
      GO TO 350                         NINPRTT 2513
C
500 SPANC = SPAN/CINCH(1)             NINPRTT 2514
ADUMPC = ADUMP/(CINCH(1)**2)        NINPRTT 2515
WRITE(6,553) NFWD,NSTA,SPANC      NINPRTT 2516
WRITE(6,555) CD,ADUMPC            NINPRTT 2518
TEM = (TTIO(1) + 460.)/1.8        NINPRTT 2519
                                         NINPRTT 2520

```

CONTENTS

	Page	
SUMMARY	1	1/A5
INTRODUCTION	1	1/A5
OVERVIEW	2	1/A6
Method of Analysis	2	1/A6
Blade geometric model	2	1/A6
Numerical model	3	1/A7
General Program Description	3	1/A7
DETAILED PROGRAM PROCEDURE	5	1/A9
Plock Data NGASDAT	6	1/A10
Main Program NTTACT	6	1/A10
Subroutine FLOWS	7	1/A11
Subroutine FLSPLT	7	1/A11
Subroutine GASTBL	8	1/A12
Subroutine GAUSS	8	1/A12
Subroutine GETIN	8	1/A12
Function HCFRCD	9	1/A13
Subroutine HCOOL	9	1/A13
Subroutine HCPINS	9	1/A13
Subroutine INPRT	9	1/A13
Subroutine PARRAY	9	1/A13
Subroutine PLNUM	10	1/A14
Subroutine PLOTMF	10	1/A14
Subroutine PREP	10	1/A14
Subroutine TARRAY	10	1/A14
Subroutine TCOEF	11	1/B1
Subroutine THRCOM	11	1/B1
Subroutine WROUT	11	1/B1
DICTIONARY	11	1/B1
PROGRAM LISTING	53	1/E1
REFERENCES	150	2/E1

```

PTIOC = PTIO(1)/CPRSR(1)          NINPRTT 2521
PEXC = PEX(ICHLNL)/CPRSR(1)      NINPRTT 2522
WPLENC = WPLEN/CMSFL(1)          NINPRTT 2523
WRITE(6,557) TEM,PTIOC,PEXC,WPLENC NINPRTT 2524
ITRBG = NFWD + 2                 NINPRTT 2525
WRITE(6,154) ICHNL,ITRBG        NINPRTT 2526
DO 518 I = 1,NSTA,20            NINPRTT 2527
IP18 = I + 18                  NINPRTT 2528
IF (IP18.GT.NSTA) IP18 = NSTA   NINPRTT 2529
IF (I.EQ.1) WRITE(6,556) (J,J=I,IP18,2) NINPRTT 2530
IF (I.GT.1) WRITE(6,559) (J,J=I,IP18,2) NINPRTT 2531
ID = 0                          NINPRTT 2532
DO 504 J = I,IP18,2            NINPRTT 2533
ID = ID + 1                    NINPRTT 2534
DUM1(ID) = RR(J)/CINCH(1)      NINPRTT 2535
NFLUID(J) = 5*j                NINPRTT 2536
504 CONTINUE
      WRITE(6,558) (NFLUID(J),J=I,IP18,2)
      ID = 0
      DO 516 J = I,IP18,2
      ID = ID + 1
      NOS = NFLUID(J) - 4
      IF (NOS.GT.1) GO TO 506
      XOS = 0.0
      XJN = 0.0
      XMM = 0.0
      XIS = 0.0
      XCC = 0.0
      GO TO 508
506 XOS = XOS + DLX(NOS)/CINCH(1) NINPRTT 2541
      XJN = XJN + DLX(NOS+1)/CINCH(1) NINPRTT 2542
      XMM = XMM + DLX(NOS+2)/CINCH(1) NINPRTT 2543
      XIS = XIS + DLX(NOS+3)/CINCH(1) NINPRTT 2544
      XCC = XCC + DLX(NOS+4)/CINCH(1) NINPRTT 2545
508 CONTINUE
      DUM2(ID) = XOS
      DUM25(ID) = XJN
      DUM3(ID) = XMM
      DUM4(ID) = XIS
      DUM5(ID) = XCC
      DUM55(ID) = TAU(NOS)/CINCH(1)
      DUM6(ID) = TAU(NOS+2)/CINCH(1)
      NOS = NFLUID(J)
      DUM7(ID) = TAU(NOS)/CINCH(1)
      DUM8(ID) = A(NOS)/(CINCH(1)**2)
      DUM9(ID) = DH(J)/CINCH(1)
      DUM10(ID) = DHJ(J)/CINCH(1)
      DUM11(ID) = CNUM(J)
      DUM12(ID) = AJET(J)/(CINCH(1)**2)
      DUM16(ID) = THUBIN(J)/1.8
      DUM17(ID) = QHUBIN(J)/CHFLX(1)
      DUM18(ID) = TTIPIN(J)/1.8
      DUM19(ID) = QTIPIN(J)/CHFLX(1)
      NOS = NFLUID(J) - 4
      DUM13(ID) = TG(J)/1.8
      DUM14(ID) = HG(J)/CHTC(1)
      JHCAL = IHC(J)
      DUM15(ID) = HCAL(JHCAL)
      IF (BT&.GT..01) DUM14(ID) = QG(J)/CHFLX(1)
      CONTINUE

```

```

      WRITE(6,560) (DUM1(J),J=1,1D)
      WRITE(6,562) (DUM2(J),J=1,1D)
      WRITE(6,563) (DUM25(J),J=1,1D)
      WRITE(6,564) (DUM3(J),J=1,1D)
      WRITE(6,566) (DUM4(J),J=1,1D)
      WRITE(6,568) (DUM5(J),J=1,1D)
      WRITE(6,569) (DUM55(J),J=1,1D)
      WRITE(6,570) (DUM6(J),J=1,1D)
      WRITE(6,572) (DUM7(J),J=1,1D)
      WRITE(6,574) (DUM8(J),J=1,1D)
      WRITE(6,576) (DUM9(J),J=1,1D)
      WRITE(6,578) (DUM10(J),J=1,1D)
      WRITE(6,580) (DUM11(J),J=1,1D)
      WRITE(6,582) (DUM12(J),J=1,1D)
      WRITE(6,583) (DUM15(J),J=1,1D)
      WRITE(6,584) (DUM13(J),J=1,1D)
      IF (BTA.LT..01) WRITE(6,586) (DUM14(J),J=1,1D)
      IF (BTA.GT..01) WRITE(6,588) (DUM14(J),J=1,1D)
      IF (ICHNL.GT.1) GO TO 518
      IF (IHUB.EQ.1) WRITE(6,596) (DUM16(J),J=1,1D)
      IF (IHUB.EQ.3) WRITE(6,598) (DUM17(J),J=1,1D)
      IF (ITIP.EQ.1) WRITE(6,602) (DUM18(J),J=1,1D)
      IF (ITIP.EQ.3) WRITE(6,604) (DUM19(J),J=1,1D)

518   CONTINUE
      ITRBG = NFWD + 1
      WRITE(6,190) ICHNL,ITRBG
      XOS = 0.0
      XJN = 0.0
      XMM = 0.0
      XIS = 0.0
      XCC = 0.0
      DO 540 I = 2,NSTA,20
      IP18 = I + 18
      IF (IP18.GT.NSTA) IP18 = STA-1
      IF (I.EQ.2) WRITE(6,556) (J,J=I,IP18,2)
      IF (I.GT.2) WRITE(6,559) (J,J=I,IP18,2)
      ID = 0
      DO 522 J = I,IP18,2
      ID = ID + 1
      DUM1(ID) = RR(J)/CINCH(1)
      NFLUID(J) = 5*J
      522   CONTINUE
      WRITE(6,558) (NFLUID(J),J=I,IP18,2)
      ID = 0
      DO 530 J = I,IP18,2
      ID = ID + 1
      NOS = NFLUID(J) - 4
      XOS = XOS + DLX(NOS)/CINCH(1)
      XJN = XJN + DLX(NOS+1)/CINCH(1)
      XMM = XMM + DLX(NOS+2)/CINCH(1)
      XIS = XIS + DLX(NOS+3)/CINCH(1)
      XCC = XCC + DLX(NOS+4)/CINCH(1)
      DUM2(ID) = XOS
      DUM25(ID) = XJN
      DUM3(ID) = XMM
      DUM4(ID) = XIS
      DUM5(ID) = XCC
      DUM55(ID) = TAU(NOS)/CINCH(1)
      DUM6(ID) = TAU(NOS+2)/CINCH(1)
      NOS = NFLUID(J)

      NINPRTT 2581
      NINPRTT 2582
      NINPRTT 2583
      NINPRTT 2584
      NINPRTT 2585
      NINPRTT 2586
      NINPRTT 2587
      NINPRTT 2588
      NINPRTT 2589
      NINPRTT 2590
      NINPRTT 2591
      NINPRTT 2592
      NINPRTT 2593
      NINPRTT 2594
      NINPRTT 2595
      NINPRTT 2596
      NINPRTT 2597
      NINPRTT 2598
      NINPRTT 2599
      NINPRTT 2600
      NINPRTT 2601
      NINPRTT 2602
      NINPRTT 2603
      NINPRTT 2604
      NINPRTT 2605
      NINPRTT 2606
      NINPRTT 2607
      NINPRTT 2608
      NINPRTT 2609
      NINPRTT 2610
      NINPRTT 2611
      NINPFTT 2612
      NINPRTT 2613
      NINPRTT 2614
      NINPRTT 2615
      NINPRTT 2616
      NINPRTT 2617
      NINPRTT 2618
      NINPRTT 2619
      NINPFTT 2620
      NINPRTT 2621
      NINPRTT 2622
      NINPRTT 2623
      NINPRTT 2624
      NINPRTT 2625
      NINPRTT 2626
      NINPRTT 2627
      NINPFTT 2628
      NINPRTT 2629
      NINPFTT 2630
      NINPRTT 2631
      NINPFTT 2632
      NINPFTT 2633
      NINPFTT 2634
      NINPRTT 2635
      NINPRTT 2636
      NINPRTT 2637
      NINPRTT 2638
      NINPRTT 2639
      NINPFTT 2640

```

```

DUM7 (ID) = TAU(NOS)/CINCH(1) NINPRTT 2641
DUM8 (ID) = A(NOS)/(CINCH(1)**2) NINPRTT 2642
DUM9 (ID) = DH(J)/CINCH(1) NINPRTT 2643
DUM10 (ID) = DHJ(J)/CINCH(1) NINPRTT 2644
DUM11 (ID) = CNUM(J) NINPRTT 2645
DUM12 (ID) = AJET(J)/(CINCH(1)**2) NINPRTT 2646
DUM16 (ID) = THUBIN(J)/1.8 NINPRTT 2647
DUM17 (ID) = QHUBIN(J)/CHFLX(1) NINPRTT 2648
DUM18 (ID) = TTIPIN(J)/1.8 NINPRTT 2649
DUM19 (ID) = QTIPIN(J)/CHFLX(1) NINPRTT 2650
NOS = NFLUID(J) - 4 NINPRTT 2651
DUM13 (ID) = TG(J)/1.8 NINPRTT 2652
DUM14 (ID) = HG(J)/CHTC(1) NINPRTT 2653
JHCAL = IHC(J) NINPRTT 2654
DUM15 (ID) = HCAL(JHCAL) NINPRTT 2655
IF (BTA.GT..01) DUM14 (ID) = QG(J)/CHFLX(1) NINPRTT 2656
530 CONTINUE NINPRTT 2657
      WRITE(6,560) (DUM1(J),J=1,ID) NINPRTT 2658
      WRITE(6,562) (DUM2(J),J=1,ID) NINPRTT 2659
      WRITE(6,563) (DUM25(J),J=1,ID) NINPRTT 2660
      WRITE(6,564) (DUM3(J),J=1,ID) NINPRTT 2661
      WRITE(6,566) (DUM4(J),J=1,ID) NINPRTT 2662
      WRITE(6,568) (DUM5(J),J=1,ID) NINPRTT 2663
      WRITE(6,569) (DUM55(J),J=1,ID) NINPRTT 2664
      WRITE(6,570) (DUM6(J),J=1,ID) NINPRTT 2665
      WRITE(6,572) (DUM7(J),J=1,ID) NINPRTT 2666
      WRITE(6,574) (DUM8(J),J=1,ID) NINPRTT 2667
      WRITE(6,576) (DUM9(J),J=1,ID) NINPRTT 2668
      WRITE(6,578) (DUM10(J),J=1,ID) NINPRTT 2669
      WRITE(6,580) (DUM11(J),J=1,ID) NINPRTT 2670
      WRITE(6,582) (DUM12(J),J=1,ID) NINPRTT 2671
      WRITE(6,583) (DUM15(J),J=1,ID) NINPRTT 2672
      WRITE(6,584) (DUM13(J),J=1,ID) NINPRTT 2673
      IF (BTA.LT..01) WRITE(6,596) (DUM14(J),J=1,ID) NINPRTT 2674
      IF (BTA.GT..01) WRITE(6,588) (DUM14(J),J=1,ID) NINPRTT 2675
      IF (ICHNL.GT.1) GO TO 540 NINPRTT 2676
      IF (IHUB.EQ.1) WRITE(6,596) (DUM16(J),J=1,ID) NINPRTT 2677
      IF (IHUB.EQ.3) WRITE(6,598) (DUM17(J),J=1,ID) NINPRTT 2678
      IF (ITIP.EQ.1) WRITE(6,602) (DUM18(J),J=1,ID) NINPRTT 2679
      IF (ITIP.EQ.3) WRITE(6,604) (DUM19(J),J=1,ID) NINPRTT 2680
540 CONTINUE NINPRTT 2681
553 FORMAT(/' NUMBER OF STATIONS IN IMPINGEMENT REGION IS',I3,
      Z      ', TOTAL NUMBER OF STATIONS IS',I3,
      Z      ', SPAN OF THIS SLICE IS',F6.3,' CM') NINPRTT 2682
555 FORMAT(' IMPINGEMENT HOLE DISCHARGE COEF.=',F6.3, NINPRTT 2683
      Z      ', AREA OF DUMP TO TRAILING EDGE =',F8.5,' CM**2') NINPRTT 2684
557 FORMAT(' COOLANT INLET TEMP.=',F7.1,' K, COOLANT INLET', NINPRTT 2685
      Z      ' PRESSURE =',F7.1,' KPA, EXIT PRESSURE =', NINPRTT 2686
      Z      F7.1,' KPA,'/ COOLANT FLOW =',F6.1,' KG/HR') NINPRTT 2687
556 FORMAT(//' STATION NUMBER',5X,10(6X,I4)) NINPRTT 2688
558 FORMAT(' COOLANT NODE NUMBER',10(6X,I4)) NINPRTT 2689
559 FORMAT(1H//' STATION NUMBER',5X,10(6X,I4)) NINPRTT 2690
560 FORMAT(' RADIAL LOCATION(CM)',10F10.3) NINPRTT 2691
562 FORMAT(' X, OUTSIDE SUR. (CM)',10F10.5) NINPRTT 2692
563 FORMAT(' X, INTERFACE (CM)',10F10.5) NINPRTT 2693
564 FORMAT(' X, MID-METAL (CM)',10F10.5) NINPRTT 2694
566 FORMAT(' X, INSIDE SURF. (CM)',10F10.5) NINPRTT 2695
568 FORMAT(' X, MID.COOL.CH.(CM)',10F10.5) NINPRTT 2696
569 FORMAT(/' COATING THKNESS (CM)',10F10.5) NINPRTT 2697
570 FORMAT(' WALL THICKNESS (CM)',10F10.5) NINPRTT 2698
                                         NINPRTT 2699
                                         NINPRTT 2700

```

```

572 FORMAT(' CHANNEL WIDTH (CM)',10F10.5) NINPRTT 2701
574 FORMAT(' CHANNEL AREA(CM**2)',10F10.5) NINPRTT 2702
576 FORMAT(' CHANNEL HYD.DIA(CM)',10F10.5) NINPRTT 2703
578 FORMAT(' IMP.JET HYD.DIA(CM)',10F10.5) NINPRTT 2704
580 FORMAT(' NO. OF IMP. JETS ',10F10.2) NINPRTT 2705
582 FORMAT(' TOT.JET AREA(CM**2)',10F10.5) NINPRTT 2706
583 FORMAT(' TYPE OF HC CALC. ',10(6X,A4)) NINPRTT 2707
584 FORMAT(' OUTSIDE BC: TG,(K)',10F10.1) NINPRTT 2708
586 FORMAT(' HG (W/M**2/K)',10F10.1) NINPRTT 2709
588 FORMAT(' QG (W/M**2)',10F10.1) NINPRTT 2710
592 FORMAT(' CLAD K (W/M/K)',10F10.3) NINPRTT 2711
594 FORMAT(' METAL K (W/M/K)',10F10.3) NINPRTT 2712
596 FORMAT(' GIVEN HUB TEMP. (K)',10F10.1) NINPRTT 2713
598 FORMAT(' QHUB (W/M**2)',10F10.1) NINPRTT 2714
602 FORMAT(' GIVEN TIP TEMP. (K)',10F10.1) NINPRTT 2715
604 FORMAT(' QTIP (W/M**2)',10F10.1) NINPRTT 2716
C NINPRTT 2717
350 CONTINUE NINPRTT 2718
RETURN NINPRTT 2719
END NINPRTT 2720

C----SOURCE.NPARAYT NPARAYT 2721
SUBROUTINE PARRAY(JS,JSENS,ICHORKE)
C NPARAYT 2722
C- SOURCE.NPARAYT---- NPARAYT 2723
C++ A SUBROUTINE TO SET UP THE COEF ARRAY TO SOLVE FOR BLADE PRESSURES NPARAYT 2724
C NPARAYT 2725
REAL*8 TCOF NPARAYT 2726
C NPARAYT 2727
COMMON /MATRIX/ TCOF(400,30) NPARAYT 2728
C NPARAYT 2729
COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),
Z CPC(80), GAMC(80), DUMR1(80), DUMR2(80) NPARAYT 2730
C NPARAYT 2731
COMMON /TCO/ ADUMP, BTA, CD, CP,
Z GAM, PIM, R, SPAN, TOG, NPARAYT 2732
Z WDUMP, WIM, AKC(15,80), AKW(15,80), NPARAYT 2733
Z A(400), AJET(80), AM2(80), CNUM(80), NPARAYT 2734
Z DH(80), DHP(80), DHJ(80), NPARAYT 2735
Z DLX(400), FF(80), HC(80), HG(80), NPARAYT 2736
Z P(2,15,80), PEXIT(15), PUMP(80), QG(80), NPARAYT 2737
Z QSNK(80), RR(80), S(15), T(2,15,400), NPARAYT 2738
Z TG(80), TAU(400), WPC(80), XN(80), NPARAYT 2739
Z WJ(15,80), WCROS(2,15,80), NPARAYT 2740
Z ICOR, IFILM, IHUB, ITIP, NPARAYT 2741
Z ISBLOK, ISLICE, NBLKSZ, NSLICE, NPARAYT 2742
Z NFWD, NSTA, IHC(80) NPARAYT 2743
C NPARAYT 2744
COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,
Z DLTME, TYME, TEPS, TYMMAX NPARAYT 2745
C NPARAYT 2746
DIMENSION POLD(80), PSAV(5) NPARAYT 2747
C NPARAYT 2748
COMPUTE NEW PRESSURES NPARAYT 2749
C NPARAYT 2750
C IPNL = THE NUMBER OF FLOW CHANNEL NODES NPARAYT 2751
C NPARAYT 2752
TREPS = 1.0 NPARAYT 2753
IF (TYME.GE.0.) TREPS = TEPS NPARAYT 2754
800 IPNL = NSTA - 3 NPARAYT 2755
NODST = 5*NSTA NPARAYT 2756
NPARAYT 2757
NPARAYT 2758
NPARAYT 2759
NPARAYT 2760

```

```

NODSP = 5*NPWD          NPARAYT 2761
C
C INITIALIZE COEFFICIENT ARRAY TO 0.0      NPARAYT 2762
C
DO 810 I = 1,IPNL        NPARAYT 2763
DO 810 J = 1,30          NPARAYT 2764
810 TCOF(I,J) = 0.0      NPARAYT 2765
C
C COMPUTE THE COEFFICIENT VALUES      NPARAYT 2766
C
DO 900 I = 1,IPNL        NPARAYT 2767
FILM = 0.0                NPARAYT 2768
820 ICHK = I - 2*(I/2)    NPARAYT 2769
C
C FOR THE IMPINGEMENT REGION:      NPARAYT 2770
ICHK = 0 IMPLIES I IS EVEN AND STATION IS ON SUCTION SIDE
= 1 IMPLIES THAT I IS ODD AND STATION IS ON PRESSURE SIDE
C
C DEFINE THE REAL NODE NUMBER IN TERMS OF I      NPARAYT 2771
C WHERE IPL IS THE PIVOTAL ELEMENT = COOLANT NODE NUMBER, LCOOL
C IDN = DOWNSTREAM COOLANT NODE      NPARAYT 2772
C IUP = UPSTREAM COOLANT NODE      NPARAYT 2773
C
IF ( I.LT.NPWD ) GO TO 840      NPARAYT 2774
IF ( I.EQ.NPWD ) GO TO 890      NPARAYT 2775
C
C FOR I=NPWD, THE NODE IS THE ENTRANCE TO THE TRAILING EDGE AND IS      NPARAYT 2776
C TREATED SEPARATELY AT (890)
C FOR I>NPWD, THE NODE IS IN THE TRAILING EDGE AND IRL IS DEFINED AS:      NPARAYT 2777
C
IF ( ICHK.GT.0 ) GO TO 885      NPARAYT 2778
IRL = 5*I      NPARAYT 2779
IDN = IRL + 10      NPARAYT 2780
IDNS = I+2      NPARAYT 2781
IUP = IRL      NPARAYT 2782
IUPS = IDNS - 2      NPARAYT 2783
ITC = 10      NPARAYT 2784
ITCP = 12      NPARAYT 2785
830 CONTINUE      NPARAYT 2786
GO TO 860      NPARAYT 2787
C
840 CONTINUE      NPARAYT 2788
IRL = 5*I      NPARAYT 2789
IF ( I.GT.JS ) GO TO 843      NPARAYT 2790
C
IF ( I.LT.JS ) GO TO 852      NPARAYT 2791
C
IF ( ICHK.GT.0 ) GO TO 849      NPARAYT 2792
C
GO TO 855      NPARAYT 2793
C
843 IF ( ICHK.GT.0 ) GO TO 849      NPARAYT 2794
C
C STATION I IS SUCTION SIDE, DOWNSTREAM OF SPLIT POINT      NPARAYT 2795
C
846 IUPS = I - 2      NPARAYT 2796
IDNS = I      NPARAYT 2797
IUP = IRL - 10      NPARAYT 2798
IDN = IRL      NPARAYT 2799
ITC = 8      NPARAYT 2800

```

```

ITCP = 10          NPARAYT 2821
IF (I.GT.2) GO TO 860   NPARAYT 2822
IUPS = 1          NPARAYT 2823
IUP = 5           NPARAYT 2824
ITC = 9            NPARAYT 2825
GO TO 860          NPARAYT 2826
C
C STATION I IS PRESSURE SIDE,DOWNSTREAM OF SPLIT POINT
C
849  CONTINUE
IUPS = I          NPARAYT 2827
IDNS = I + 2       NPARAYT 2828
IUP = IRL          NPARAYT 2829
IDN = IRL + 10    NPARAYT 2830
ITC = 10           NPARAYT 2831
ITCP = 12           NPARAYT 2832
GO TO 860          NPARAYT 2833
852  CONTINUE
IF (ICHK.GT.0) GO TO 858   NPARAYT 2834
IF (ICHK.NE.JSENS) GO TO 846  NPARAYT 2835
855  CONTINUE
C
C I IS ON SUCTION SIDE, FORWARD OF SPLIT POINT
C
IUPS = I          NPARAYT 2836
IDNS = I - 2       NPARAYT 2837
IUP = IRL          NPARAYT 2838
IDN = IRL - 10    NPARAYT 2839
IDX = IUP          NPARAYT 2840
ITC = 10           NPARAYT 2841
ITCP = 8            NPARAYT 2842
IF (I.GT.2) GO TO 860   NPARAYT 2843
IDNS = 1            NPARAYT 2844
IDN = 5             NPARAYT 2845
ITCP = 9            NPARAYT 2846
GO TO 860          NPARAYT 2847
858  CONTINUE
IF (ICHK.NE.JSENS) GO TO 849  NPARAYT 2848
IDNS = I            NPARAYT 2849
IUPS = I + 2       NPARAYT 2850
IDN = IRL          NPARAYT 2851
IUP = IRL + 10    NPARAYT 2852
IDX = IUP          NPARAYT 2853
ITC = 12           NPARAYT 2854
ITCP = 10           NPARAYT 2855
GO TO 860          NPARAYT 2856
860  CONTINUE
C
C
TRTRM = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRM = 12.*DLX(IDN)*
Z (WCROS(2,ISLICE,IDNS)-WCROS(1,ISLICE,IDNS))/(DLTYME*A(IUP)*32.2)  NPARAYT 2857
WFCDUM = WFC(IDNS)          NPARAYT 2858
IF (I.GT.NFWD) WFCDUM = WFCDUM + WFC(IDNS+1)  NPARAYT 2859
IF (WCROS(2,ISLICE,IDNS).NE.0.0) FILM = WFCDUM/WCROS(2,ISLICE,IDNS)  NPARAYT 2860
TCOP(I,ITC) = TREPS*  NPARAYT 2861
Z ((1.0 + GAMC(IUPS)*AM2(IUPS)) + (A(IDN)-A(IUP))/(2.*A(IUP)))  NPARAYT 2862
TCOF(I,ITCP) = TREPS*(-(1.0 + .5*GAMC(IDNS)*AM2(IDNS)*  NPARAYT 2863
Z (4.*FF(IDNS)*DLX(IDX)/DH(IDNS)+2.+2.*FILM))*A(IDN)/A(IUP)  NPARAYT 2864
Z + (A(IDN)-A(IUP))/(2.*A(IUP)))  NPARAYT 2865

```

```

ROOT      = SQRT(32.2*GAMC(TDNS)*R*T(2,ISLICE, IDN)*AM2(IDNS))    NPARAYT 2881
PUMTRM = 0.0
IF (ROOT.NE.0.0) PUMTRM = (3.14159265*WS/30.)**2*
Z   RR(IDNS)*(RR(IDNS)-RR(IUPS))*WCROS(2,ISLICE, IDNS)          NPARAYT 2882
Z   /(A(IUP)*ROOT*144.*32.2)                                         NPARAYT 2883
TCOF(I,20) = -PUMTRM + TRTRM - (1.-TREPS)*                           NPARAYT 2884
Z   (P(1,ISLICE,IUPS)*TCOF(I,ITC) + P(1,ISLICE, IDNS)*TCOF(I,ITCP)) NPARAYT 2885
970  CONTINUE
IF (IDNS.NE.ICHOKE) GO TO 880                                         NPARAYT 2886
TCOF(I,20) = -P(1,ISLICE,ICHOKE)*TCOP(I,12) + TCOP(I,20)             NPARAYT 2887
TCOF(I,12) = 0.0                                                       NPARAYT 2888
880  CONTINUE
C
C FOR TRAILING EDGE CHANNELS:
C
IF (I.LT.IFNL) GO TO 900
C
C TCOF(I,20) IS NON-ZERO ONLY FOR I=IFNL
C
IF (ICHOKE.EQ.NSTA-1) GO TO 900
TCOF(I,20) = -PEXIT(ISLICE)*TCOF(I,12) + TCOP(I,20)
TCOF(I,12) = 0.0
GO TO 900
885  CONTINUE
C
C FOR A PRESSURE SIDE, TRAILING EDGE REGION STATION, COOLANT NODE
C IS IDENTICAL TO SUCTION SIDE NODE.
C
TCOF(I,10) = 1.0
TCOF(I,9) = -1.0
TCOF(I,20) = 0.0
GO TO 900
890  CONTINUE
C
C FOR THE SPECIAL NODE AT THE ENTRANCE TO THE TRAILING EDGE:
C ALLOWING FOR THE POSSIBILITY OF ADDITION OF EXTRA COOLING AIR
C INTO TRAILING EDGE,
C
TRTRM = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRM=12.*DLX(NFWD+1)*
Z   (WCROS(2,ISLICE,NFWD+1)-WCROS(1,ISLICE,NFWD+1))/           NPARAYT 2900
Z   (DLTYME*A(NODSF+5)*32.2)                                         NPARAYT 2901
AVRGA = (A(NODSF-5) + A(NODSF) - A(NODSF+5))/(3.*A(NODSF+5))    NPARAYT 2902
TCOF(I,9) = TREPS*((1. + GAMC(NFWD-1)*AM2(NFWD-1))*              NPARAYT 2903
Z   A(NODSF-5)/A(NODSF+5) - AVRGA)                                     NPARAYT 2904
TCOF(I,10) = TREPS*((1. + GAMC(NFWD)*AM2(NFWD))*                  NPARAYT 2905
Z   A(NODSF)/A(NODSF+5) - AVRGA)                                     NPARAYT 2906
IF (WCROS(2,ISLICE,NFWD+1).NE.0.0) FILM =
Z   (WFC(NFWD+1)+WFC(NFWD+2))/WCROS(2,ISLICE,NFWD+1)                NPARAYT 2907
TCOF(I,11) = TREPS*(-1. - GAMC(NFWD+1)*AM2(NFWD+1))*               NPARAYT 2908
Z   (1. + 2.*FF(NFWD+1)*DLX(NODSF+5)/DH(NFWD+1)+FILM) - AVRGA)   NPARAYT 2909
C
PUMP(NFWD+1) = (3.14159265*WS/30.)**2*                                NPARAYT 2910
Z   RR(NFWD+1)*(RR(NFWD+1)-RR(NFWD))                                    NPARAYT 2911
ROOT      = SQRT(32.2*GAMC(NFWD+1)*R*                                NPARAYT 2912
Z   T(2,ISLICE,NODSF+5)*AM2(NFWD+1))                                     NPARAYT 2913
PUMTRM = 0.0
IF (ROOT.NE.0.0) PUMTRM = (3.14159265*WS/30.)**2*RR(NFWD+1)*        NPARAYT 2914
Z   (RR(NFWD+1)-RR(NFWD))*WCROS(2,ISLICE,NFWD+1)                      NPARAYT 2915

```

```

Z           /(A(NODSP+5)*ROOT*144.*32.2)          NPARAYT 2941
C
C PUMP HAS UNITS OF (IN**2/SEC**2);   ROOT HAS UNITS OF (FT/SEC)  NPARAYT 2942
C
C DUMTER = 0.0          NPARAYT 2943
C IF (ADUMP.GT.0.) DUMTER = - WDUMP**2*R*          NPARAYT 2944
Z           (T(2,ISLICE,NODSP-5)+T(2,ISLICE,NODSP))/          NPARAYT 2945
Z ((P(1,ISLICE,NFWD-1) + P(1,ISLICE,NFWD))*ADUMP*A(NODSP+5)*32.2)  NPARAYT 2946
Z TCOF(I,20) = - PUMTRM + DUMTER + TRTRM          NPARAYT 2947
Z - (1.0-TREPS)*(P(1,ISLICE,NFWD-1)*TCOF(I,9)          NPARAYT 2948
Z + P(1,ISLICE,NFWD)*TCOF(I,10)+P(1,ISLICE,NFWD+1)*TCOF(I,11))  NPARAYT 2949
900    CONTINUE          NPARAYT 2950
C
C RETURN          NPARAYT 2951
C END          NPARAYT 2952
C
C----SOURCE.NPLENMP          NPLENMP 2953
C SUBROUTINE PLNUM(WXX,PXX,PTEXIT,TXX,TTEXIT)          NPLENMP 2954
C
C SOURCE.NPLENMP----          NPLENMP 2955
C A SUBROUTINE TO COMPUTE PRESSURE DROP IN THE CENTRAL COOLANT PLENUM          NPLENMP 2956
C
C
C
C
C
C -----
C ARGUMENTS FOR THIS SUBROUTINE ARE:          NPLENMP 2957
C
C WXX      = MASS FLOW RATE INTO THIS SLICE, LBM/HR          NPLENMP 2958
C PXX      = AVERAGE STATIC PRESSURE FOR THIS PLENUM SLICE, PSIA,          NPLENMP 2959
C           CALCULATED IN THIS SUBROUTINE.          NPLENMP 2960
C PTEXIT = TOTAL PRESSURE IN AND TOTAL PRESSURE OUT, PSIA          NPLENMP 2961
C TXX      = AVERAGE STATIC TEMPERATURE FOR THIS PLENUM SLICE, (F),          NPLENMP 2962
C           CALCULATED IN THIS SUBROUTINE.          NPLENMP 2963
C TTEXIT = TOTAL TEMP. IN AND TOTAL TEMP.OUT, (F)          NPLENMP 2964
C
C -----
C COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),          NPLENMP 2965
C Z          PIN(15), TIN(15), W(15), WS          NPLENMP 2966
C
C COMMON /TCO/ ADUMP, BTA, CD, CP,          NPLENMP 2967
C Z          GAM, PIM, R, SPAN, TOG,          NPLENMP 2968
C Z          WDUMP, WIM, AKC(15,80), AKW(15,80),          NPLENMP 2969
C Z          A(400), AJET(80), AM2(80), CNUM(80),          NPLENMP 2970
C Z          DH(80), DHF(80), DHJ(80),          NPLENMP 2971
C Z          DLX(400), FF(80), HC(80), HG(80),          NPLENMP 2972
C Z          P(2,15,80), PEXIT(15), PUMP(80), QG(80),          NPLENMP 2973
C Z          QSNK(80), RR(80), S(15), T(2,15,400),          NPLENMP 2974
C Z          TG(80), TAU(400), WFC(80),          NPLENMP 2975
C Z          WJ(15,80), WCROS(2,15,80), XN(80),          NPLENMP 2976
C Z          ICOR, IPILM, IRUB, ITIP,          NPLENMP 2977
C Z          ISBLOK, ISLICE, NBLKSZ, NSLICE,          NPLENMP 2978
C Z          NFWD, NSTA, IHC(80)          NPLENMP 2979
C
C COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,          NPLENMP 2980
C Z          DLTYME, TYME, TEPS, TYMMAX          NPLENMP 2981
C
C COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),          NPLENMP 2982
C

```

```

Z          CTMPP (2) , CTCOM (2) , CDEN (2) , CSPHT (2) , CGASC (2) ,
Z          CVISC (2) , CRHOVG (2) , IUNITS                         NPLENMP 3001
C
C          DIMENSION BETTA (20) , B (20) , AMG (20) , SIGMA (20) , TT1 (20) , P1 (20) ,
1          Z1 (15) , CH (15)                                         NPLENMP 3002
C          DIMENSION SV (3) , XK (4) , XL (4)                         NPLENMP 3003
C
C          FUNP (PP) IS THE EQUATION FOR DELTA P OVER LENGTH DX      NPLENMP 3004
C
C          FUNP (PP) = (D1*RRP* (PP/R/TP-V1)/144.0-2.*PP*G2*TP/PP/AA/AA/DD)
Z          / (1.-TP*G2/(PP*AA)**2+G1*CP)                           NPLENMP 3005
Z          *778.161*R*TP*TP*V1/(PP*AA)**2/PP)*DX                   NPLENMP 3006
C
C          FUNT (XK) IS THE EQUATION FOR DELTA T OVER LENGTH DX      NPLENMP 3007
C
C          FUNT (XK) = (D2*RRP/CP+G1*R*(TP/PP/AA)**2*(XK/PP/DX))    NPLENMP 3008
Z          /(1.+G1*R*TP/(PP*AA)**2)*DX                            NPLENMP 3009
C
C          INITIALIZE                                              NPLENMP 3010
1          CONTINUE                                               NPLENMP 3011
DIFTOL=0.005
ACH=1.
KSIG=0
NCC=1
IS=0
KTR1=0
W(ISLICE) = WXX
C
C          SAVE INLET TOTAL PRESSURE (PSIA) IN PIN AND INLET TOTAL    NPLENMP 3012
C          TEMPERATURE (F) IN TIN                                     NPLENMP 3013
C
C          PIN (ISLICE) = PTEXIT                                     NPLENMP 3014
C          TIN (ISLICE) = TTEXIT                                     NPLENMP 3015
C          CH (ISLICE) = 0.0                                       NPLENMP 3016
C          ZED=Z1 (ISLICE)*1.01                                     NPLENMP 3017
C          IF (ZED.EQ.0.) ZED=.001                                 NPLENMP 3018
C          IF (TIN (ISLICE).GT.-430.0) GO TO 5                  NPLENMP 3019
3          TIN (ISLICE)=50.0                                     NPLENMP 3020
5          SIGB=0.0                                           NPLENMP 3021
C
C          NSTNS= 4                                                 NPLENMP 3022
C          SEGMTS=NSTNS-1                                         NPLENMP 3023
C
C          T1=TIN (ISLICE)+460.0                                    NPLENMP 3024
C          B (1)=T1                                              NPLENMP 3025
C          BETTA1=PIN (ISLICE)**2                                NPLENMP 3026
C          BETTA (1)=BETTA1                                     NPLENMP 3027
C          DX=S (ISLICE)/SEGMTS                               NPLENMP 3028
C          DXTEMP=DX                                         NPLENMP 3029
C          XXN=NSTNS                                         NPLENMP 3030
C
C          DR= (ROUT (ISLICE)-RIN (ISLICE))/SEGMTS             NPLENMP 3031
C
C          COMPUTE CONSTANT TERMS-C1-C8 -
C
17         CONTINUE                                              NPLENMP 3032
TTX=B (1)                                         NPLENMP 3033

```

```

CALL GASTBL(TTX,C,CP,GAM,PD,R,XMU) NPLENMP 3061
J=1 NPLENMP 3062
C6=.5*(GAM-1.0) NPLENMP 3063
C1=GAM/C6 NPLENMP 3064
C NPLENMP 3065
C NPLENMP 3066
C IF (WS) 21,19,21 NPLENMP 3067
C NO PUMPING NPLENMP 3068
19 C3=0.0 NPLENMP 3069
GO TO 23 NPLENMP 3070
C NPLENMP 3071
C PUMPING NPLENMP 3072
C NPLENMP 3073
21 C3=2.36695E-6*(WS**2)/(C1*R) NPLENMP 3074
23 C8=32.17*GAM*R NPLENMP 3075
C5=1.0/SQRT(C8) NPLENMP 3076
C7=1.0/(32.17*C1*R) NPLENMP 3077
IF (J.GT.1) GO TO 33 NPLENMP 3078
25 CONTINUE NPLENMP 3079
C NPLENMP 3080
C COMPUTE CHANNEL REYNOLDS NO. IF J = 1 NPLENMP 3081
C REY = 12.0*W(ISLICE)/XMU*DPLN(ISLICE)/APLN(ISLICE) NPLENMP 3082
C NPLENMP 3083
C COMPUTE FRICTION FACTOR NPLENMP 3084
C NPLENMP 3085
C COMPUTE Z TERMS NPLENMP 3086
33 CONTINUE NPLENMP 3087
Z3=12.0*W(ISLICE)/XMU NPLENMP 3088
Z4=(R*W(ISLICE)/3600.0)**2 NPLENMP 3089
IF (J.GT.1) GO TO 77 NPLENMP 3090
C DETERMINE INLET CONDITIONS NPLENMP 3091
35 CONTINUE NPLENMP 3092
NPLENMP 3093
NPLENMP 3094
NPLENMP 3095
C INITIAL STATION COMPUTATIONS -
NPLENMP 3096
C BALANCING ON TOTAL PRESSURE -
NPLENMP 3097
NPLENMP 3098
39 NAG=-1 NPLENMP 3099
41 SIGC=(B(J)/APLN(ISLICE))**2*Z4/BETTA(J) NPLENMP 3100
IF (ABS(SIGB-SIGC).LE..00001*SIGC) GO TO 57 NPLENMP 3101
C SIGMA NOT CONVERGED NPLENMP 3102
43 IS=IS+1 NPLENMP 3103
SV(IS)=SIGC NPLENMP 3104
IF (IS.EQ.3) GO TO 135 NPLENMP 3105
45 B(J)=T1-C7*SIGC NPLENMP 3106
IF (B(J).LT.50.0) GO TO 159 NPLENMP 3107
C TEMP OK NPLENMP 3108
47 SIGB=SIGC NPLENMP 3109
BETTA(J)=BETA1*(B(J)/T1)**C1 NPLENMP 3110
GO TO 41 NPLENMP 3111
NPLENMP 3112
C SIGMA CONVERGED NPLENMP 3113
57 B(J)=T1-C7*SIGC NPLENMP 3114
AMC(1)=SQRT(SIGC/B(J))*C5 NPLENMP 3115
IF (B(J).LE.0.0) GO TO 159 NPLENMP 3116
KTPBZ=0 NPLENMP 3117
63 BETTA(J)=BETA1/(1.0+C6*AMC(1)**2)**C1 NPLENMP 3118
IF (BETTA(J).LE.0.) GO TO 159 NPLENMP 3119
65 B(1)=T1/(1.0+C6*AMC(1)**2) NPLENMP 3120

```

```

SIGMA(1) = (B(J)/APLN(ISLICE))**2*Z4/BETTA(J) NPLENMP 3121
SIGC=SQRT(SIGMA(1)/B(1))*C5 NPLENMP 3122
IF (ABS(SIGC-AMC(1)).LE..01) GO TO 71 NPLENMP 3123
67 AMC(1)=SIGC NPLENMP 3124
69 KTRBZ=KTRBZ+1 NPLENMP 3125
IF (KTRBZ.LE.20) GO TO 63 NPLENMP 3126
71 TT1(1)=TIN(ISLICE) NPLENMP 3127
NPLENMP 3128
NPLENMP 3129
NPLENMP 3130
NPLENMP 3131
NPLENMP 3132
NPLENMP 3133
NPLENMP 3134
NPLENMP 3135
NPLENMP 3136
NPLENMP 3137
NPLENMP 3138
NPLENMP 3139
NPLENMP 3140
NPLENMP 3141
NPLENMP 3142
NPLENMP 3143
NPLENMP 3144
NPLENMP 3145
NPLENMP 3146
NPLENMP 3147
NPLENMP 3148
NPLENMP 3149
NPLENMP 3150
NPLENMP 3151
NPLENMP 3152
NPLENMP 3153
NPLENMP 3154
NPLENMP 3155
NPLENMP 3156
NPLENMP 3157
NPLENMP 3158
NPLENMP 3159
NPLENMP 3160
NPLENMP 3161
NPLENMP 3162
NPLENMP 3163
NPLENMP 3164
NPLENMP 3165
NPLENMP 3166
NPLENMP 3167
81 MACH1=XNN NPLENMP 3168
WRITE(6,83) ISLICE,J NPLENMP 3169
83 FORMAT(5X,9H***** ,50HDECREASED INCREMENT DERIVATIVE CHANGING NPLENMP 3170
1 TOO FAST,3X,'BRANCH NO. ',I2,', STATION NO. ',I2/) NPLENMP 3171
XNN=XNN*2.0 NPLENMP 3172
DX=DX/2.0 NPLENMP 3173
PTEMP=BETTA(J-1) NPLENMP 3174
TTEMP=B(J-1) NPLENMP 3175
RTP =RTEMP NPLENMP 3176
DD= DPLN(ISLICE) NPLENMP 3177
AA= APLN(ISLICE) NPLENMP 3178
DO 91 L=1,4 NPLENMP 3179
V1=G1/PP/AA**2/(1.0+G1*R*TP/(PP*AA)**2) NPLENMP 3180
TERM1=TP*G2/(PP*AA)**2

```

```

TERM2=G1*CP*778.161*R*TP*TP*V1/(PP*AA)**2/PP          NPLENMP 3181
TESTMA=1.0-TERM1+TERM2        NPLENMP 3182
IF (TESTMA.LE.0.0) GO TO 159      NPLENMP 3183
XK(L)=FUNP(F1(J))      NPLENMP 3184
IF (L.EQ.1) GO TO 89      NPLENMP 3185
DO 87 LL=2,L      NPLENMP 3186
XTEST=ABS((XK(L)-XK(LL-1))/PP)      NPLENMP 3187
IF (XTEST.GT.DIFTOL) GO TO 81      NPLENMP 3188
87 CONTINUE      NPLENMP 3189
89 XL(L)=PUNT(XK(L))      NPLENMP 3190
IF (L.EQ.4) GO TO 93      NPLENMP 3191
PP =PTEMP+XK(L)/2.0      NPLENMP 3192
TP =TTEMP+XL(L)/2.0      NPLENMP 3193
IF (L.EQ.2) GO TO 91      NPLENMP 3194
RRP=RRP+DR/XNN      NPLENMP 3195
IF (L.NE.3) GO TO 91      NPLENMP 3196
PP =PTEMP+XK(L)      NPLENMP 3197
TP =TTEMP+XL(L)      NPLENMP 3198
91 CONTINUE      NPLENMP 3199
93 PP =PTEMP+XK(L)      NPLENMP 3200
TP =TTEMP+XL(L)      NPLENMP 3201
IF (PP.LE.0.0.OR.TP.LE.0.0) GO TO 159      NPLENMP 3202
V1=G1/PP/AA**2/(1.0+G1*R*TP/(PP*AA)**2)      NPLENMP 3203
TERM1=TP*G2/(PP*AA)**2      NPLENMP 3204
TERM2=G1*CP*778.161*R*TP*TP*V1/(PP*AA)**2/PP          NPLENMP 3205
TESTMA=1.0-TERM1+TERM2        NPLENMP 3206
IF (TESTMA.LE.0.0) GO TO 159      NPLENMP 3207
BETTA(J)=PTEMP+(XK(1)+2.0*(XK(2)+XK(3))+XK(4))/6.0      NPLENMP 3208
B(J)=TTEMP+(XL(1)+2.0*(XL(2)+XL(3))+XL(4))/6.0      NPLENMP 3209
PP = BETTA(J)      NPLENMP 3210
IF (B(J).LE.0.0.OR.BETTA(J).LE.0.0) GO TO 159      NPLENMP 3211
TP =B(J)      NPLENMP 3212
IF (MACH1.EQ.1) GO TO 95      NPLENMP 3213
MACH1=MACH1-1      NPLENMP 3214
PTEMP = PP      NPLENMP 3215
TTEMP = TP      NPLENMP 3216
GO TO 85      NPLENMP 3217
95 XNN=2.0      NPLENMP 3218
DX=DXTEMP      NPLENMP 3219
SIGMA(J)=B(J)/APLN(ISLICE)/BETTA(J)*SQRT(Z4)      NPLENMP 3220
AMC(J)=SIGMA(J)*SQRT(1.0/B(J))*C5      NPLENMP 3221
IF (AMC(J).GE.1.0) GO TO 159      NPLENMP 3222
F1(J)=REY      NPLENMP 3223
TT1(J)=B(J)*(1.0+C6*AMC(J)**2)-460.0      NPLENMP 3224
97 CONTINUE      NPLENMP 3225
C ALL STATIONS COMPUTED      NPLENMP 3226
99 SIGC = 1.0      NPLENMP 3227
AMC(NSTNS)=AMC(NSTNS)/SIGC      NPLENMP 3228
IF (AMC(NSTNS).GT.1.0) GO TO 159      NPLENMP 3229
BETTA(NSTNS)=BETTA(NSTNS)*SIGC**2      NPLENMP 3230
C RESTART CHOKED BRANCH IF M.LT. .8      NPLENMP 3231
IF (CH(ISLICE).EQ.0..OR.CH(ISLICE).EQ.1.) GO TO 113      NPLENMP 3232
IF (ACH.EQ.(-1.0)) GO TO 113      NPLENMP 3233
ACH=-1.      NPLENMP 3234
AB=0.      NPLENMP 3235
DO 109 J=1,NSTNS      NPLENMP 3236
IF (AMC(J).GT..8) GO TO 113      NPLENMP 3237
IF (AMC(J).LT.AB) GO TO 109      NPLENMP 3238
AB=AMC(J)      NPLENMP 3239
109 CONTINUE      NPLENMP 3240

```

```

CH(ISLICE)=0.0 NPLENMP 3241
AJ=(GAM+1.)/2. NPLENMP 3242
CX=-(GAM+1.0)/(GAM-1.0)/2.0 NPLENMP 3243
AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX) NPLENMP 3244
WCHOKE=W(ISLICE) NPLENMP 3245
W(ISLICE)=AZ*W(ISLICE) NPLENMP 3246
WRITE(6,111) ISLICE,W(ISLICE),WCHOKE NPLENMP 3247
111 FORMAT(8X,6H*****,23H RESTART CHOKED BRANCH ,I5,24H FLOW RATE INC NPLENMP 3248
REASED TO ,F10.4,6H FROM ,F10.4,7H *****) NPLENMP 3249
GO TO 177 NPLENMP 3250
113 BETA1=PIN(ISLICE)**2 NPLENMP 3251
C CALCULATE THE CHOKING FLOW RATE NPLENMP 3252
AB=0.0 NPLENMP 3253
DO 115 J=1,NSTNS NPLENMP 3254
IF (AMC(J).LT.AB) GO TO 115 NPLENMP 3255
AB=AMC(J) NPLENMP 3256
115 CONTINUE NPLENMP 3257
AJ=(GAM+1.0)/2.0 NPLENMP 3258
CX=-(GAM+1.0)/(GAM-1.0)/2.0 NPLENMP 3259
AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX) NPLENMP 3260
C COMPUTE RESISTANCE EQUATION FOR BALANCE NPLENMP 3261
PT1=(1.0+C6*AMC(1)**2)**(C1/2.0)*BETTA(1) NPLENMP 3262
IF (C3.NE.0.0) GO TO 117 NPLENMP 3263
GO TO 119 NPLENMP 3264
117 DR2=ROUT(ISLICE)**2-RIN(ISLICE)**2 NPLENMP 3265
119 PEXTT=PT1*(1.0+C3/T1*DR2)**(C1/2.0) NPLENMP 3266
121 CONTINUE NPLENMP 3267
Z1(ISLICE)=(PEXTT**2-BETTA(NSTNS)**2)/W(ISLICE)**2 NPLENMP 3268
IF (Z1(ISLICE).GT.0.0) GO TO 129 NPLENMP 3269
WRITE(6,125) ISLICE,Z1(ISLICE) NPLENMP 3270
125 FORMAT(//5X,'PASSAGE ',I3,5X,'HAS NEGATIVE OR NO RFSISTANCE' NPLENMP 3271
      ,F12.4//) NPLENMP 3272
Z1(ISLICE)=ZED NPLENMP 3273
129 CONTINUE NPLENMP 3274
PP = BETTA(NSTNS)*(1.+C6*AMC(NSTNS)**2)**(C1/2.0) NPLENMP 3275
PTEXIT = PP NPLENMP 3276
DIFTOL=0.005 NPLENMP 3277
KTR1=0 NPLENMP 3278
C COMPUTE AVERAGE STATIC PRESSURE AND STATIC TEMPERATURE NPLENMP 3279
C NPLENMP 3280
PXX = 0.0 NPLENMP 3281
TXX = 0.0 NPLENMP 3282
DO 134 I = 1,NSTNS NPLENMP 3283
TXX = TXX + B(I) NPLENMP 3284
134 PXX = PXX + BETTA(I) NPLENMP 3285
TXX = TXX/XXN - 460. NPLENMP 3286
PXX = PXX/XXN NPLENMP 3287
TTEXIT = TT1(NSTNS) NPLENMP 3288
RETURN NPLENMP 3289
C COMPUTE ACCELERATION NPLENMP 3290
C NPLENMP 3291
135 D=SV(2)-SV(1) NPLENMP 3292
137 D=(SV(3)-SV(2))/D NPLENMP 3293
E=ABS(D)-1.0 NPLENMP 3294
IF (ABS(D).GT..6) GO TO 143 NPLENMP 3295
139 E=D/(D-1.0) NPLENMP 3296
141 SIGB=E*SV(2)+(1.0-E)*SV(3) NPLENMP 3297
143 IS=0 NPLENMP 3298
NPLENMP 3299
NPLENMP 3300

```

```

145  SIGC=SIGB          NPLENMP 3301
    IF (SIGC.LE.0.) GO TO 159      NPLENMP 3302
147  KSIG=KSIG+1          NPLENMP 3303
    IF (KSIG.LT.50) GO TO 155      NPLENMP 3304
149  WRITE (6,151) ISLICE,J      NPLENMP 3305
151  FORMAT(7X,28HPROGRAM IS LOOPING IN BRANCH,I6,9H STATION,I4//)
    KSIG=0          NPLENMP 3306
    NCC=NCC+1          NPLENMP 3307
    IF (NCC.LT.4) GO TO 159      NPLENMP 3308
153  ISLICE=220          NPLENMP 3309
    GO TO 129          NPLENMP 3310
155  CONTINUE          NPLENMP 3311
    IF (NAG.LT.0) GO TO 45      NPLENMP 3312
157  STOP          NPLENMP 3313
C          NPLENMP 3314
C          NPLENMP 3315
C          CCHOKING ADJUSTMENT
C          NPLENMP 3316
C          NPLENMP 3317
159  WCHOKE=W(ISLICE)      NPLENMP 3318
    DIFTOL=0.050          NPLENMP 3319
161  IF (CH(ISLICE).EQ.0.) GO TO 165      NPLENMP 3320
163  W(ISLICE)=.98*W(ISLICE)      NPLENMP 3321
    IF (KTR1.GT.20) W(ISLICE)=W(ISLICE)*.98** (KTR1-20)
    GO TO 171          NPLENMP 3322
165  AB=0.          NPLENMP 3323
    IF (J.EQ.1) GO TO 169          NPLENMP 3324
    DO 167 IJK=1,J          NPLENMP 3325
    IF (AMC(IJK).GT.1.0) AMC(IJK)=1.0      NPLENMP 3326
    IF (AMC(IJK).LT.AB) GO TO 167      NPLENMP 3327
    AB=AMC(IJK)          NPLENMP 3328
167  CONTINUE          NPLENMP 3329
    AJ=(GAM+1.)/2.          NPLENMP 3330
    CX=-(GAM+1.0)/(GAM-1.0)/2.0      NPLENMP 3331
    AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)
    W(ISLICE)=AZ*W(ISLICE)      NPLENMP 3332
    GO TO 171          NPLENMP 3333
169  CONTINUE          NPLENMP 3334
    W(ISLICE)=1600.0*APLN(ISLICE)*SQRT((32.17*BETA1*GAM)/(R*T1))
171  CH(ISLICE)=1.0          NPLENMP 3335
173  WRITE(6,175) ISLICE,J,W(ISLICE),WCHOKE      NPLENMP 3336
175  FORMAT(3X,12H*** PASSAGE ,I5,23H HAS CHOKED AT STATION ,I5,
    134H AND THE FLOW HAS BEEN REDUCED TO ,F10.4,6H FROM ,F10.4,4H ***)
    IS=0          NPLENMP 3337
177  BETTA(1)=PIN(ISLICE)**2      NPLENMP 3338
    B(1)=T1          NPLENMP 3339
    DX=DXTTEMP      NPLENMP 3340
    J=1          NPLENMP 3341
    SIGB=0.0          NPLENMP 3342
    KTF1=KTR1+1      NPLENMP 3343
    IF (KTR1.GE.50) GO TO 181      NPLENMP 3344
179  GO TO 17          NPLENMP 3345
181  WRITE (6,183) ISLICE      NPLENMP 3346
183  FORMAT(/2X,16H**FLOW IN BRANCH,I6,
    Z      ' HAS BEEN REDUCED 50 TIMES BECAUSE OF CHOKING')
    GO TO 153          NPLENMP 3347
    END          NPLENMP 3348
                                NPLENMP 3349
                                NPLENMP 3350
                                NPLENMP 3351
                                NPLENMP 3352
                                NPLENMP 3353
                                NPLENMP 3354
                                NPLENMP 3355

```

```

C----SOURCE.NPLOTIT          NPLOTIT 3356
    SUBROUTINE PLOTMF(ALPH2)      NPLOTIT 3357
C          NPLOTIT 3358
C          SOURCE.NPLOTIT          NPLOTIT 3359
C          NPLOTIT 3360

```

```

COMMON /PRPS/ CPO,      GAMO,      DP(80),      SP(80),      RE(80),
Z           CPC(80),    GAMC(80),    DUMR1(80),   DUMR2(80)          NPLOTIT 3361
NPLOTIT 3362
NPLOTIT 3363
NPLOTIT 3364
NPLOTIT 3365
NPLOTIT 3366
NPLOTIT 3367
NPLOTIT 3368
NPLOTIT 3369
NPLOTIT 3370
NPLOTIT 3371
NPLOTIT 3372
NPLOTIT 3373
NPLOTIT 3374
NPLOTIT 3375
NPLOTIT 3376
NPLOTIT 3377
NPLOTIT 3378
NPLOTIT 3379
NPLOTIT 3380
NPLOTIT 3381
NPLOTIT 3382
NPLOTIT 3383
NPLOTIT 3384
NPLOTIT 3385
NPLOTIT 3386
NPLOTIT 3387
NPLOTIT 3388
NPLOTIT 3389
NPLOTIT 3390
NPLOTIT 3391
NPLOTIT 3392
NPLOTIT 3393
NPLOTIT 3394
NPLOTIT 3395
NPLOTIT 3396
NPLOTIT 3397
NPLOTIT 3398
NPLOTIT 3399
NPLOTIT 3400
NPLOTIT 3401
NPLOTIT 3402
NPLOTIT 3403
NPLOTIT 3404
NPLOTIT 3405
NPLOTIT 3406
NPLOTIT 3407
NPLOTIT 3408
NPLOTIT 3409
NPLOTIT 3410
NPLOTIT 3411
NPLOTIT 3412
NPLOTIT 3413
NPLOTIT 3414
NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),
Z           IPLOT, MD1, MD2, MD3, IADJIN, IWRITE          NPLOTIT 3364
NPLOTIT 3365
NPLOTIT 3366
NPLOTIT 3367
NPLOTIT 3368
NPLOTIT 3369
NPLOTIT 3370
NPLOTIT 3371
NPLOTIT 3372
NPLOTIT 3373
NPLOTIT 3374
NPLOTIT 3375
NPLOTIT 3376
NPLOTIT 3377
NPLOTIT 3378
NPLOTIT 3379
NPLOTIT 3380
NPLOTIT 3381
NPLOTIT 3382
NPLOTIT 3383
NPLOTIT 3384
NPLOTIT 3385
NPLOTIT 3386
NPLOTIT 3387
NPLOTIT 3388
NPLOTIT 3389
NPLOTIT 3390
NPLOTIT 3391
NPLOTIT 3392
NPLOTIT 3393
NPLOTIT 3394
NPLOTIT 3395
NPLOTIT 3396
NPLOTIT 3397
NPLOTIT 3398
NPLOTIT 3399
NPLOTIT 3400
NPLOTIT 3401
NPLOTIT 3402
NPLOTIT 3403
NPLOTIT 3404
NPLOTIT 3405
NPLOTIT 3406
NPLOTIT 3407
NPLOTIT 3408
NPLOTIT 3409
NPLOTIT 3410
NPLOTIT 3411
NPLOTIT 3412
NPLOTIT 3413
NPLOTIT 3414
NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

COMMON /TCO/ ADUME,      PTA,      CD,      CP,
Z           GAM,      PIM,      R,      SPAN,      TOG,
Z           WDUMP,    WIM,      AKC(15,80), AKW(15,80),
Z           A(400),    AJET(80), AM2(80),  CNUM(80),
Z           DH(80),    DHF(80), DHJ(80),
Z           DLX(400),  FP(80),  HC(80),  HG(80),
Z           P(2,15,80), PEXIT(15), PUMP(80), QG(80),
Z           QSNK(80),  RR(80),  S(15),   T(2,15,400),
Z           FG(80),    TAU(400), WFC(80),
Z           WJ(15,80), WCROS(2,15,80), XN(80),
Z           ICOR,     IFILM,   IHUB,   ITIF,
Z           ISBLOK,   ISLICE,  NBLKSZ, NSLICE,
Z           NFWD,     NSTA,    IHC(80)          NPLOTIT 3367
NPLOTIT 3368
NPLOTIT 3369
NPLOTIT 3370
NPLOTIT 3371
NPLOTIT 3372
NPLOTIT 3373
NPLOTIT 3374
NPLOTIT 3375
NPLOTIT 3376
NPLOTIT 3377
NPLOTIT 3378
NPLOTIT 3379
NPLOTIT 3379
NPLOTIT 3380
NPLOTIT 3381
NPLOTIT 3382
NPLOTIT 3383
NPLOTIT 3384
NPLOTIT 3385
NPLOTIT 3386
NPLOTIT 3387
NPLOTIT 3388
NPLOTIT 3389
NPLOTIT 3390
NPLOTIT 3391
NPLOTIT 3392
NPLOTIT 3393
NPLOTIT 3394
NPLOTIT 3395
NPLOTIT 3396
NPLOTIT 3397
NPLOTIT 3398
NPLOTIT 3399
NPLOTIT 3400
NPLOTIT 3401
NPLOTIT 3402
NPLOTIT 3403
NPLOTIT 3404
NPLOTIT 3405
NPLOTIT 3406
NPLOTIT 3407
NPLOTIT 3408
NPLOTIT 3409
NPLOTIT 3410
NPLOTIT 3411
NPLOTIT 3412
NPLOTIT 3413
NPLOTIT 3414
NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

COMMON /TRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,
Z           DLTYME,    TYME,      TEPS,      TYMAX          NPLOTIT 3381
NPLOTIT 3382
NPLOTIT 3383
NPLOTIT 3384
NPLOTIT 3385
NPLOTIT 3386
NPLOTIT 3387
NPLOTIT 3388
NPLOTIT 3389
NPLOTIT 3390
NPLOTIT 3391
NPLOTIT 3392
NPLOTIT 3393
NPLOTIT 3394
NPLOTIT 3395
NPLOTIT 3396
NPLOTIT 3397
NPLOTIT 3398
NPLOTIT 3399
NPLOTIT 3400
NPLOTIT 3401
NPLOTIT 3402
NPLOTIT 3403
NPLOTIT 3404
NPLOTIT 3405
NPLOTIT 3406
NPLOTIT 3407
NPLOTIT 3408
NPLOTIT 3409
NPLOTIT 3410
NPLOTIT 3411
NPLOTIT 3412
NPLOTIT 3413
NPLOTIT 3414
NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSPL(2),
Z           CTMPF(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2),
Z           CVISC(2), CRHOVG(2), IUNITS          NPLOTIT 3384
NPLOTIT 3385
NPLOTIT 3386
NPLOTIT 3387
NPLOTIT 3388
NPLOTIT 3389
NPLOTIT 3390
NPLOTIT 3391
NPLOTIT 3392
NPLOTIT 3393
NPLOTIT 3394
NPLOTIT 3395
NPLOTIT 3396
NPLOTIT 3397
NPLOTIT 3398
NPLOTIT 3399
NPLOTIT 3400
NPLOTIT 3401
NPLOTIT 3402
NPLOTIT 3403
NPLOTIT 3404
NPLOTIT 3405
NPLOTIT 3406
NPLOTIT 3407
NPLOTIT 3408
NPLOTIT 3409
NPLOTIT 3410
NPLOTIT 3411
NPLOTIT 3412
NPLOTIT 3413
NPLOTIT 3414
NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

DIMENSION Y(320), XLBL(29), YLBL(7), TLBL1(21), TLBL2(9)
DIMENSION XLBL2(15), VARIB(15), YLBL(10), ALPH2(4), ALABL(7)
DIMENSION XS(80), XP(80), TSO(500), TSM(500), TPO(500), TPM(500)
DIMENSION YLBL2(11)
DIMENSION PLEGN(5), SLEGN(5), SYMBL(10), YLBL(20), XLBL(20)
DIMENSION YTEM(80), ISYM(5), PLTYME(2)
LOGICAL*1 IXAX/.TRUE./, IYAX/.FALSE./
INTEGER*2 NPTS
DIMENSION RTNARR(2), VARS(12), IVARS(12)
DATA PLEGN/'A' , 'PRES', 'SURE', 'SID', 'E'   '/
DATA SLEGN/'B' , 'SUCT', 'ION', 'SID', 'E'   '/
DATA XLBL2/15*' '
DATA SYMBOL/'0'/
DATA XLBL/'MID', 'WALL', 'X/L', 'PRO', 'M ST', 'ATIO', 'N NO', . 1, 'NPLOTIT 3401
Z           , 'L =', ' ', ' ', 'INC', 'HES', ' ', ' ', ' ', 'NPLOTIT 3402
Z           , ' ', 'SID', 'E; C', 'HNL', ' ', ' ', 'PL', 'ENUM', 'PRE', 'NPLOTIT 3403
Z           , 'SS.=', ' ', ' ', 'PSI', 'A***', 'NPLOTIT 3404
DATA YLBL/-----, 'TEMP', 'ERAT', 'URE', 'DEG', 'F.', -----/
DATA YLBL/ CO, 'OLAN', 'T CH', 'ANNE', 'L T', 'OTAL', 'PRE'
Z           , 'SSUR', 'E', 'PSIA'/          NPLOTIT 3405
DATA YLBL2/7*' ', 'TEMP', 'ERAT', 'URE', 'F'  /
DATA TLBL1/21*' '
DATA TLBL2/9*' '
DATA VARIB/'PRES', 'SURE', 'SUC', 'TION', 'SURF', 'ACE', 'MID',
Z           'WALL', 'CM', ' ', 'KPA', '****', 'K', 'KP', 'A'  /
DATA ALABL/2*' ', '----', '2*', '----', ' ', 'NPLOTIT 3413
DATA SYMBL/1', '2', '3', '4', '5', '6', '7', '8', '9', '0'/
DATA ISYM/62, 119, 118, 70, 65/
DATA PLTYME/2*' '
ATYME = TYME
IF (ATYME.LT.0.0) ATYME = 0.0
CALL NUMBER(4, ATYME, 8, 2, PLTYME)
ALABL(6) = PLTYME(1)          NPLOTIT 3414
NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

```

```

      ALABL(7) = PLTYME(2)          NPLOTIT 3421
C   C MD2 > 0 INDICATES JOB IS COMPLETE.--NOW DO SUMMARY PLOTS.  NPLOTIT 3422
C
10    CONTINUE          NPLOTIT 3423
     IF (MD2.GT.0) GO TO 80        NPLOTIT 3424
20    NSTAPS = NSTA/2 + 1         NPLOTIT 3425
     NLBLS = NSTAPS/5            NPLOTIT 3426
C
C   SET UP TIME AND DATE LABEL FOR PLOT IDENTIFICATION  NPLOTIT 3427
C
      ALABL(1) = ALPH2(3)          NPLOTIT 3428
      ALABL(2) = ALPH2(4)          NPLOTIT 3429
      ALABL(3) = ALPH2(1)          NPLOTIT 3430
      ALABL(4) = ALPH2(2)          NPLOTIT 3431
C
C   SET UP TITLE          NPLOTIT 3432
C
      DO 45 I = 1,30              NPLOTIT 3433
      IF (I.LE.21) TLBL1(I) = TITLE(I)
      IF (I.GT.21) TLBL2(I-21) = TITLE(I)
45    CONTINUE          NPLOTIT 3434
C
C   PRESSURE SIDE          NPLOTIT 3435
C
46    IF (MD3.GT.1) GO TO 55        NPLOTIT 3436
47    XP(1) = 0.0                  NPLOTIT 3437
     IX = 1                      NPLOTIT 3438
     DO 50 I = 3,NSTA,2           NPLOTIT 3439
C
C   NMM IS THE MIDMETAL NODE NUMBER (L)          NPLOTIT 3440
C
      NMM = 5*I - 2              NPLOTIT 3441
     IX = IX + 1                NPLOTIT 3442
50    XP(IX) = XP(IX-1) + DLX(NMM)/CINCH(IUNITS)  NPLOTIT 3443
     XPL = XP(NSTAPS)            NPLOTIT 3444
     DO 51 I = 2,NSTAPS          NPLOTIT 3445
51    XP(I) = XP(I)/XPL          NPLOTIT 3446
55    CONTINUE          NPLOTIT 3447
     IY = 0                      NPLOTIT 3448
     ITP = NSTAPS*(ISLICE-1)      NPLOTIT 3449
     DO 60 I = 1,NSTA,2           NPLOTIT 3450
     IY = IY + 1                NPLOTIT 3451
     NOS = 5*I - 4              NPLOTIT 3452
     Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)  NPLOTIT 3453
     IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
     ITP = ITP + 1              NPLOTIT 3454
     TPO(ITP) = Y(IY)            NPLOTIT 3455
     IYP = IY + NSTAPS          NPLOTIT 3456
     Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)  NPLOTIT 3457
     IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
     IYP = IY + 2*NSTAPS          NPLOTIT 3458
     Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)  NPLOTIT 3459
     IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
     TPM(ITP) = Y(IYP)            NPLOTIT 3460
     IYP = IY + 3*NSTAPS          NPLOTIT 3461
     Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)  NPLOTIT 3462
     IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
     IYP = IY + 4*NSTAPS          NPLOTIT 3463
     NCOOL = NOS + 4             NPLOTIT 3464

```

```

Y(IYP) = T(2,ISLICE,NCOOL)/CTMPP(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
XLBL(16) = VARIB(1)
XLBL(17) = VARIB(2)
IF (IUNITS.EQ.2) GO TO 601
XLBL(12) = VARIB(9)
XLBL(13) = VARIB(10)
XLBL(28) = VARIB(11)
XLBL(29) = VARIB(12)
YLBL(5) = VARIB(13)
YLBL(6) = VARIB(10)
YLBL(10) = VARIB(11)
YLBL2(11) = VARIB(13)
601 CONTINUE
DO 611 I = 1,15
611 XLBL2(I) = XLBL(I+15)
C
IF (IPLOT.EQ.3) GO TO 63
C
CPIM = PIM/CPRSR(IUNITS)
CALL NUMBER(1,ISLICE,4,0,XLBL2(6))
CALL NUMBER(4,CPIM,8,1,XLBL2(11))
CALL NUMBER(4,XPL,8,4,XLBL(10))
CALL CHARS(84,TLBL1,0.0,0.15,9.85,12)
CALL CHARS(36,TLBL2,0.0,0.15,9.65,12)
CALL CHAES(60,XLBL,0.0,1.5,.25,12)
CALL CHARS(56,XLBL2,0.0,1.5,.05,12)
CALL CHARS(28,YLBL,90.,.25,3.3,12)
MD3 = MD3 + 1
CALL NUMEER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,6.2,9.3,12)
C
C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS
C
NPTS = NSTAPS
CALL SCALE(IXAX,NPTS,XP)
NPTS = 5*NSTAPS
CALL SCLBAK(IYAX,NPTS,Y,RTNAFR)
CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
VARS(1) = 7.0
VARS(2) = 9.0
VARS(3) = 0.0
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
CALL XAXIS(.8,.6,VARS)
VARS(2) = 8.9
VARS(3) = 90.
VARS(4) = YMIN
VARS(5) = YMAX
CALL YAXIS(.8,.6,VARS)
C
C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING
C DIFFERENT SYMBOLS FOR EACH.
C
IVARS(1) = 4
IVARS(2) = NSTAPS
IVARS(3) = 2
DO 603 I = 1,5
NPLOTIT 3481
NPLOTIT 3482
NPLOTIT 3483
NPLOTIT 3484
NPLOTIT 3485
NPLOTIT 3486
NPLOTIT 3487
NPLOTIT 3488
NPLOTIT 3489
NPLOTIT 3490
NPLOTIT 3491
NPLOTIT 3492
NPLOTIT 3493
NPLOTIT 3494
NPLOTIT 3495
NPLOTIT 3496
NPLOTIT 3497
NPLOTIT 3498
NPLOTIT 3499
NPLOTIT 3500
NPLOTIT 3501
NPLOTIT 3502
NPLOTIT 3503
NPLOTIT 3504
NPLOTIT 3505
NPLOTIT 3506
NPLOTIT 3507
NPLOTIT 3508
NPLOTIT 3509
NPLOTIT 3510
NPLOTIT 3511
NPLOTIT 3512
NPLOTIT 3513
NPLOTIT 3514
NPLOTIT 3515
NPLOTIT 3516
NPLOTIT 3517
NPLOTIT 3518
NPLOTIT 3519
NPLOTIT 3520
NPLOTIT 3521
NPLOTIT 3522
NPLOTIT 3523
NPLOTIT 3524
NPLOTIT 3525
NPLOTIT 3526
NPLOTIT 3527
NPLOTIT 3528
NPLOTIT 3529
NPLOTIT 3530
NPLOTIT 3531
NPLOTIT 3532
NPLOTIT 3533
NPLOTIT 3534
NPLOTIT 3535
NPLOTIT 3536
NPLOTIT 3537
NPLOTIT 3538
NPLOTIT 3539
NPLOTIT 3540

```

```

IVARS(4) = ISYM(I)
IYST = 1 + (I-1)*NSTAPS
IYEN = I*NSTAPS
III = 0
DO 602 II = IYST,IYEN
III = III + 1
602 YTEM(III) = Y(II)
CALL GPLOT(XP,YTEM,IVARS)
603 CONTINUE
CALL DISPLA(1)
C
C
C PRESSURE SIDE COOLANT PRESSURE DISTRIBUTION
C
IY = 0
DO 61 I = 1,NSTA,2
IY = IY + 1
61 Y(IY) = P(2,ISLICE,I)*
Z (1.+(GAMC(I)-1.)*AM2(I)/2.)**(GAMC(I)/(GAMC(I)-1.))/CPASR(IUNITS) NPLOTIT 3559
CALL CHARS(84,TLABL1,0.0,0.15,9.85,12) NPLOTIT 3560
CALL CHARS(36,TLABL2,0.0,0.15,9.65,12) NPLOTIT 3561
CALL CHARS(60,XLABL,0.0,1.5,.25,12) NPLOTIT 3562
CALL CHARS(56,XLABL2,0.0,1.5,.05,12) NPLOTIT 3563
CALL CHARS(40,YPLABL,90.,.25,2.8,12) NPLOTIT 3564
MD3 = MD3 + 1 NPLOTIT 3565
CALL NUMBER(1,MD3,4,0,ALABL(5)) NPLOTIT 3566
CALL CHAES(28,ALABL,C.0,6.2,9.3,12) NPLOTIT 3567
NPTS = NSTAPS NPLOTIT 3568
CALL SCALE(IXAX,NPTS,XP) NPLOTIT 3569
CALL SCLBAK(IYAX,NPTS,Y,RTNARR) NPLOTIT 3570
CALL GINTVL( RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
VARS(1) = 7.0 NPLOTIT 3572
VARS(2) = 9.0 NPLOTIT 3573
VARS(3) = 0.0 NPLOTIT 3574
VARS(4) = 0.0 NPLOTIT 3575
VARS(5) = 1.0 NPLOTIT 3576
VARS(6) = .5 NPLOTIT 3577
VARS(7) = 1.0 NPLOTIT 3578
CALL XAXIS(.8,.6,VARS) NPLOTIT 3579
VARS(2) = 8.9 NPLOTIT 3580
VARS(3) = 90. NPLOTIT 3581
VARS(4) = YMIN NPLOTIT 3582
VARS(5) = YMAX NPLOTIT 3583
CALL YAXIS(.8,.6,VARS) NPLOTIT 3584
C
C--- AXES ARE SET, NOW PLOT THE PRESSURE
C
IVARS(1) = 4 NPLOTIT 3585
IVARS(2) = NSTAPS NPLOTIT 3586
IVARS(3) = 2 NPLOTIT 3587
IVARS(4) = 65 NPLOTIT 3588
CALL GPLOT(XP,Y,IVARS) NPLOTIT 3589
CALL DISPLA(1) NPLOTIT 3590
C
C SUCTION SIDE
C
IF (MD3.GT.2) GO TO 69 NPLOTIT 3591
63 XS(1) = 0.0 NPLOTIT 3592
IX = 1 NPLOTIT 3593
DO 65 I = 2,NSTA,2 NPLOTIT 3594
NPLOTIT 3595
NPLOTIT 3596
NPLOTIT 3597
NPLOTIT 3598
NPLOTIT 3599
NPLOTIT 3600

```

```

NMM = 5*I - 2
IX = IX + 1
XS(IX) = XS(IX-1) + DLX(NMM)/CINCH(IUNITS)
XSL = XS(NSTAPS)
DO 66 I = 2,NSTAPS
XS(I) = XS(I)/XSL
CONTINUE
C
IY = 1
ITS = NSTAPS*(ISLICE-1) + 1
Y(IY) = T(2,ISLICE,1)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
TSO(ITS) = Y(IY)
IYP = IY + NSTAPS
Y(IYP) = T(2,ISLICE,2)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,3)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
TSM(ITS) = Y(IYP)
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,4)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,5)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
DO 70 I = 2,NSTA,2
IY = IY + 1
NOS = 5*I - 4
Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
ITS = ITS + 1
TSO(ITS) = Y(IY)
IYP = IY + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
TSM(ITS) = Y(IYP)
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
NCCOL = NOS + 4
Y(IYP) = T(2,ISLICE,NCOOL)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
XLABL2(1) = VARIB(3)
XLABL2(2) = VARIB(4)
C
IF (IPLOT.EQ.3) GO TO 80
C
CALL NUMBER(1,ISLICE,4,0,XLBL2(6))
CALL NUMBER(4,CPIM,8,1,XLBL2(11))
CALL NUMBER(4,XSL,8,4,XLBL(10))
IX = 2 + NFWD/2
CALL CHARS(84,TLBL1,0.0,0.15,9.85,12)
CALL CHARS(36,TLBL2,0.0,0.15,9.65,12)
CALL CHARS(60,XLBL,0.0,1.5,.25,12)
CALL CHARS(56,XLBL2,0.0,1.5,.05,12)

NPLOTIT 3601
NPLOTIT 3602
NPLOTIT 3603
NPLOTIT 3604
NPLOTIT 3605
NPLOTIT 3606
NPLOTIT 3607
NPLOTIT 3608
NPLOTIT 3609
NPLOTIT 3610
NPLOTIT 3611
NPLOTIT 3612
NPLOTIT 3613
NPLOTIT 3614
NPLOTIT 3615
NPLOTIT 3616
NPLOTIT 3617
NPLOTIT 3618
NPLOTIT 3619
NPLOTIT 3620
NPLOTIT 3621
NPLOTIT 3622
NPLOTIT 3623
NPLOTIT 3624
NPLOTIT 3625
NPLOTIT 3626
NPLOTIT 3627
NPLOTIT 3628
NPLOTIT 3629
NPLOTIT 3630
NPLOTIT 3631
NPLOTIT 3632
NPLOTIT 3633
NPLOTIT 3634
NPLOTIT 3635
NPLOTIT 3636
NPLOTIT 3637
NPLOTIT 3638
NPLOTIT 3639
NPLOTIT 3640
NPLOTIT 3641
NPLOTIT 3642
NPLOTIT 3643
NPLOTIT 3644
NPLOTIT 3645
NPLOTIT 3646
NPLOTIT 3647
NPLOTIT 3648
NPLOTIT 3649
NPLOTIT 3650
NPLOTIT 3651
NPLOTIT 3652
NPLOTIT 3653
NPLOTIT 3654
NPLOTIT 3655
NPLOTIT 3656
NPLOTIT 3657
NPLOTIT 3658
NPLOTIT 3659
NPLOTIT 3660

```

```

CALL CHARS(28,YLABL,90.,.25,3.3,12) NPLOTIT 3661
MD3 = MD3 + 1 NPLOTIT 3662
CALL NUMBER(1,MD3,4,0,ALABL(5)) NPLOTIT 3663
CALL CHARS(28,ALABL,0.0,.6,2,9.3,12) NPLOTIT 3664
C NPLOTIT 3665
C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS NPLOTIT 3666
C NPLOTIT 3667
NPTS = NSTAPS NPLOTIT 3668
CALL SCALE(IYAX,NPTS,XS) NPLOTIT 3669
NPTS = 5*NSTAPS NPLOTIT 3670
CALL SCLBAK(IYAX,NPTS,Y,RTNARR) NPLOTIT 3671
CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX) NPLOTIT 3672
VARS(1) = 7.0 NPLOTIT 3673
VARS(2) = 9.0 NPLOTIT 3674
VARS(3) = 0.0 NPLOTIT 3675
VARS(4) = 0.0 NPLOTIT 3676
VARS(5) = 1.0 NPLOTIT 3677
VARS(6) = .5 NPLOTIT 3678
VARS(7) = 1.0 NPLOTIT 3679
CALL XAXIS(.8,.6,VARS) NPLOTIT 3680
VARS(2) = 8.9 NPLOTIT 3681
VARS(3) = 90. NPLOTIT 3682
VARS(4) = YMIN NPLOTIT 3683
VARS(5) = YMAX NPLOTIT 3684
CALL YAXIS(.8,.6,VARS) NPLOTIT 3685
C NPLOTIT 3686
C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING NPLOTIT 3687
C DIFFERENT SYMBOLS FOR EACH. NPLOTIT 3688
C NPLOTIT 3689
IVARS(1) = 4 NPLOTIT 3690
IVARS(2) = NSTAPS NPLOTIT 3691
IVARS(3) = 2 NPLOTIT 3692
DO 703 I = 1,5 NPLOTIT 3693
IVARS(4) = ISYM(I) NPLOTIT 3694
IYST = 1 + (I-1)*NSTAPS NPLOTIT 3695
IYEN = I*NSTAPS NPLOTIT 3696
III = 0 NPLOTIT 3697
DO 702 II = IYST,IYEN NPLOTIT 3698
III = III + 1 NPLOTIT 3699
702 YTEM(III) = Y(II) NPLOTIT 3700
CALL GPLOT(XS,YTEM,IVARS) NPLOTIT 3701
703 CONTINUE NPLOTIT 3702
CALL DISPLA(1) NPLOTIT 3703
C NPLOTIT 3704
C NPLOTIT 3705
C SUCTION SIDE COOLANT PRESSURE DISTRIBUTION NPLOTIT 3706
C NPLOTIT 3707
Y(1) = P(2,ISLICE,1)* NPLOTIT 3708
Z (1.+(GAMC(1)-1.)*AM2(1)/2.)** (GAMC(1)/(GAMC(1)-1.))/CPRSR(IUNITS) NPLOTIT 3709
IY = 1 NPLOTIT 3710
DO 75 I = 2,NSTA,2 NPLOTIT 3711
IY = IY + 1 NPLOTIT 3712
75 Y(IY) = P(2,ISLICE,I)*(1.+(GAMC(I)-1.)* NPLOTIT 3713
Z *AM2(I)/2.)** (GAMC(I)/(GAMC(I)-1.))/CPRSR(IUNITS) NPLOTIT 3714
CALL CHARS(84,TLBL1,0.0,0.15,9.85,12) NPLOTIT 3715
CALL CHARS(36,TLBL2,0.0,0.15,9.65,12) NPLOTIT 3716
CALL CHARS(60,XLBL1,0.0,1.5,.25,12) NPLOTIT 3717
CALL CHARS(56,XLBL2,0.0,1.5,.05,12) NPLOTIT 3718
CALL CHARS(40,YLBL1,90.,.25,2.8,12) NPLOTIT 3719
MD3 = MD3 + 1 NPLOTIT 3720

```

```

CALL NUMBER(1,MD3,4,0,ALABL(5))          NPLOTIT 3721
CALL CHARS(28,ALABL,0.0,6.2,9.3,12)      NPLOTIT 3722
NPTS = NSTAPS                            NPLOTIT 3723
CALL SCALE(IYAX,NPTS,XS)                 NPLOTIT 3724
CALL SCLBAK(IYAX,NPTS,Y,RTNARR)          NPLOTIT 3725
CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
VARS(1) = 7.0                             NPLOTIT 3726
VARS(2) = 9.0                             NPLOTIT 3727
VARS(3) = 0.0                             NPLOTIT 3728
VARS(4) = 0.0                             NPLOTIT 3729
VARS(5) = 1.0                             NPLOTIT 3730
VARS(6) = .5                            NPLOTIT 3731
VARS(7) = 1.0                            NPLOTIT 3732
CALL XAXIS(.8,.6,VARS)                   NPLOTIT 3733
VARS(2) = 8.9                            NPLOTIT 3734
VARS(3) = 90.                            NPLOTIT 3735
VARS(4) = YMIN                           NPLOTIT 3736
VARS(5) = YMAX                           NPLOTIT 3737
CALL YAXIS(.8,.6,VARS)                   NPLOTIT 3738
NPLOTIT 3739
C
C--- AXES ARE SET, NOW PLOT THE PRESSURE
C
IVARS(1) = 4                            NPLOTIT 3740
IVARS(2) = NSTAPS                      NPLOTIT 3741
IVARS(3) = 2                            NPLOTIT 3742
IVARS(4) = 65                           NPLOTIT 3743
CALL GPLOT(XS,Y,IVARS)                  NPLOTIT 3744
CALL DISPLA(1)                         NPLOTIT 3745
GO TO 150                                NPLOTIT 3746
NPLOTIT 3747
C
C THE FOLLOWING SECTION PUTS OUT PLOTS CONTAINING TEMPERATURES FROM
C ALL SLICES ON ONE FRAME
C
CONTINUE
IF (ISLICE.LT.NSLICE) GO TO 150
NPLOTIT 3748
NPLOTIT 3749
NPLOTIT 3750
NPLOTIT 3751
NPLOTIT 3752
NPLOTIT 3753
NPLOTIT 3754
NPLOTIT 3755
NPLOTIT 3756
NPLOTIT 3757
NPLOTIT 3758
NPLOTIT 3759
NPLOTIT 3760
NPLOTIT 3761
NPLOTIT 3762
NPLOTIT 3763
NPLOTIT 3764
NPLOTIT 3765
NPLOTIT 3766
NPLOTIT 3767
NPLOTIT 3768
NPLOTIT 3769
NPLOTIT 3770
NPLOTIT 3771
NPLOTIT 3772
NPLOTIT 3773
NPLOTIT 3774
NPLOTIT 3775
NPLOTIT 3776
NPLOTIT 3777
NPLOTIT 3778
NPLOTIT 3779
NPLOTIT 3780
C
THE FOLLOWING PUTS TWO PLOTS ON ONE FRAME OF FILM
C
FIRST PLOT THE OUTSIDE SURFACE TEMPERATURES FOR EACH
C SLICE ON THE SAME PLOT
C
NPTS = NSTAPS*NSLICE
CALL SCLBAK(IYAX,NPTS,TPC,RTNARR)
TMAXP = RTNARR(2)
TMINP = RTNARR(1)
CALL SCLBAK(IYAX,NPTS,TSO,RTNARR)
TMAXS = RTNARR(2)
TMINS = RTNARR(1)
IF (TMAXS.GT.TMAXP) TMAXP = TMAXS
IF (TMINS.LT.TMINP) TMINP = TMINS
NINTRV = (TMAXP-TMINP)/100. + 2
CALL GINTVL(TMINP,TMAXP,NINTRV,0,ATMINP,ATMAXP)
AINTRV = NINTRV
CALL CHAFS(84,TLBL1,0.0,0.15,9.85,12)
CALL CHAFS(36,TLBL2,0.0,0.15,9.65,12)
YLBL2(5) = VARIB(5)
YLBL2(6) = VARIB(6)
CALL CHARS(44,YLBL2,90.,.25,1.6,12)
VARS(1) = 8.
VARS(2) = 8.5

```

```

VARS(3) = 0.          NPLOTIT 3781
VARS(4) = 0.0         NPLOTIT 3782
VARS(5) = 1.0         NPLOTIT 3783
VARS(6) = .5          NPLOTIT 3784
VARS(7) = 1.0         NPLOTIT 3785
VARS(8) = 0.0         NPLOTIT 3786
CALL XAXIS(1.2,.5,VARS) NPLOTIT 3787
MD3 = MD3 + 1        NPLOTIT 3788
CALL NUMBER(1,MD3,4,0,ALABL(5)) NPLOTIT 3789
CALL CHARS(28,ALABL,0.0,1.3,9.5,12) NFLOTIT 3790
CALL CHARS(20,PLEGN,0.0,6.0,9.5,12) NPLOTIT 3791
VARS(1) = 7.          NPLOTIT 3792
VARS(2) = 3.8         NPLOTIT 3793
VARS(3) = 90.         NPLOTIT 3794
VARS(4) = ATMINP     NPLOTIT 3795
VARS(5) = ATMAXP     NPLOTIT 3796
VARS(6) = AINTRV     NPLOTIT 3797
VARS(7) = 1.          NPLOTIT 3798
CALL XAXIS(1.2,.5,VARS) NPLOTIT 3799
DO 100 I = 1,NSLICE   NPLOTIT 3800
JST = NSTAPS*(I-1)    NPLOTIT 3801
DO 95 J = 1,NSTAPS   NPLOTIT 3802
95  Y(J) = TPO(JST+J) NPLOTIT 3803
SYMBOL = SYMBL(I)    NPLOTIT 3804
KS = 0                NPLOTIT 3805
C
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C IDENTIFY THE CURVES.
KSTART = I + 1        NPLOTIT 3806
DO 98 K = KSTART,NSTAPS,10 NPLOTIT 3807
KS = KS + 1            NPLOTIT 3808
98  XLBL(KS) = XP(K)   NPLOTIT 3809
YLBL(KS) = Y(K)        NPLOTIT 3810
IVARS(1) = 6            NPLOTIT 3811
IVARS(2) = NLBLS        NPLOTIT 3812
IVARS(3) = 3            NPLOTIT 3813
IVARS(4) = 240+I        NPLOTIT 3814
IVARS(5) = 1            NPLOTIT 3815
IVARS(6) = 8            NPLOTIT 3816
CALL GPLOT(XLBL,YLBL,IVARS) NPLOTIT 3817
IVARS(1) = 3            NPLOTIT 3818
IVARS(2) = NSTAPS       NPLOTIT 3819
IVARS(3) = 0            NPLOTIT 3820
100 CALL GPLOT(XP,Y,IVARS) NPLOTIT 3821
C
C
VARS(1) = 7.          NPLOTIT 3822
VARS(2) = 8.5          NPLOTIT 3823
VARS(3) = 0.          NPLOTIT 3824
VARS(4) = 0.0          NPLOTIT 3825
VARS(5) = 1.0          NPLOTIT 3826
VARS(6) = .5          NPLOTIT 3827
VARS(7) = 1.0          NPLOTIT 3828
CALL XAXIS(1.2,.5,VARS) NPLOTIT 3829
CALL CHARS(42,XLBL,0.0,3.0,.05,12) NPLOTIT 3830
VARS(1) = 7.          NPLOTIT 3831
VARS(2) = 3.8          NPLOTIT 3832
VARS(3) = 90.          NPLOTIT 3833
VARS(4) = ATMINP      NPLOTIT 3834
VARS(5) = ATMAXP      NPLOTIT 3835

```

```

VAES(6) = AINTRV          N PLOTIT 3841
VARS(7) = 1.                N PLOTIT 3842
CALL YAXIS(1.2,.5,VARS)      N PLOTIT 3843
CALL CHARS(20,SLEGN,0.0,6.0,4.5,12)  N PLOTIT 3844
DO 110 I = 1,NSLICE        N PLOTIT 3845
JST = NSTAPS*(I-1)          N PLOTIT 3846
DO 105 J = 1,NSTAPS        N PLOTIT 3847
105   Y(J) = TSO(JST+J)      N PLOTIT 3848
SYMBOL = SYMBL(I)          N PLOTIT 3849
KS = 0                      N PLOTIT 3850
C
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C IDENTIFY THE CURVES.          N PLOTIT 3851
KSTART = I + 1              N PLOTIT 3852
DO 108 K = KSTART,NSTAPS,10  N PLOTIT 3853
KS = KS + 1                 N PLOTIT 3854
XLBL(KS) = XS(K)            N PLOTIT 3855
108   YLBL(KS) = Y(K)          N PLOTIT 3856
IVARS(1) = 6                 N PLOTIT 3857
IVARS(2) = NLBLS             N PLOTIT 3858
IVARS(3) = 3                 N PLOTIT 3859
IVARS(4) = 240+I             N PLOTIT 3860
IVARS(5) = 1                 N PLOTIT 3861
IVARS(6) = 8                 N PLOTIT 3862
CALL GPLOT(XLBL,YLBL,IVARS)  N PLOTIT 3863
IVARS(1) = 3                 N PLOTIT 3864
IVARS(2) = NSTAPS            N PLOTIT 3865
IVARS(3) = 0                 N PLOTIT 3866
110   CALL GPLOT(XS,Y,IVARS)  N PLOTIT 3867
CALL DISPLA(1)               N PLOTIT 3868
C
C
C NOW PLOT THE MID-METAL TEMPERATURES FOR EACH SLICE, ALL ON ONE PLOT
C
112   CONTINUE
CALL SCLBAK(IYAX,NPTS,TPM,RTNARR)  N PLOTIT 3871
TMAXP = RTNARR(2)                  N PLOTIT 3872
TMINP = RTNARR(1)                  N PLOTIT 3873
CALL SCLBAK(IYAX,NPTS,TSM,RTNARR)  N PLOTIT 3874
TMAXS = RTNARR(2)                  N PLOTIT 3875
TMINS = RTNARR(1)                  N PLOTIT 3876
IF (TMAXS.GT.TMAXP) TMAXP = TMAXS  N PLOTIT 3877
IF (TMINS.LT.TMINP) TMINE = TMINS  N PLOTIT 3878
NINTRV = (TMAXP-TMINP)/100. + 2    N PLOTIT 3879
CALL GINTVL(TMINP,TMAXP,NINTRV,0,ATMINP,ATMAXP)
AINTRV = NINTRV
CALL CHAFS(84,TLABL1,0.0,0.15,9.85,12)  N PLOTIT 3880
CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)  N PLOTIT 3881
YLBL2(5) = VARIB(7)                N PLOTIT 3882
YLBL2(6) = VARIB(8)                N PLOTIT 3883
CALL CHARS(44,YLBL2,90.,.25,1.6,12)  N PLOTIT 3884
VARS(1) = 8.                      N PLOTIT 3885
VARS(2) = 8.5                     N PLOTIT 3886
VARS(3) = 0.                      N PLOTIT 3887
VARS(4) = 0.0                     N PLOTIT 3888
VARS(5) = 1.0                     N PLOTIT 3889
VARS(6) = .5                      N PLOTIT 3890
VARS(7) = 1.0                     N PLOTIT 3891
VARS(8) = 0.0                     N PLOTIT 3892

```

```

CALL XAXIS(1.2,5.5,VARS)          NPLOTIT 3901
MD3 = MD3 + 1                    NPLOTIT 3902
CALL NUMBER(1,MD3,4,0,ALABL(5))   NPLOTIT 3903
CALL CHARS(28,ALABL,0.0,1.3,9.5,12) NPLOTIT 3904
CALL CHARS(20,PLEGN,0.0,6.0,9.5,12) NPLOTIT 3905
VARS(1) = 7.                      NPLOTIT 3906
VARS(2) = 3.8                     NPLOTIT 3907
VARS(3) = 90.                     NPLOTIT 3908
VARS(4) = ATMINP                 NPLOTIT 3909
VARS(5) = ATMAXP                 NPLOTIT 3910
VARS(6) = AINTRV                 NPLOTIT 3911
VARS(7) = 1.                      NPLOTIT 3912
CALL YAXIS(1.2,5.5,VARS)          NPLOTIT 3913
114 CONTINUE                      NPLOTIT 3914
DO 120 I = 1,NSLICE               NPLOTIT 3915
JST = NSTAPS*(I-1)
DO 115 J = 1,NSTAPS              NPLOTIT 3916
115 Y(J) = TPM(JST+J)             NPLOTIT 3917
SYMBOL = SYMBL(I)                NPLOTIT 3918
KS = 0                            NPLOTIT 3919
C
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C IDENTIFY THE CURVES.           NPLOTIT 3920
KSTART = I + 1                  NPLOTIT 3921
DO 118 K = KSTART,NSTAPS,10     NPLOTIT 3922
KS = KS + 1                     NPLOTIT 3923
XLBL(KS) = XP(K)                NPLOTIT 3924
118 YLBL(KS) = Y(K)              NPLOTIT 3925
IVAPS(1) = 6                     NPLOTIT 3926
IVARS(2) = NLBLS                NPLOTIT 3927
IVARS(3) = 3                     NPLOTIT 3928
IVARS(4) = 240+I                NPLOTIT 3929
IVARS(5) = 1                     NPLOTIT 3930
IVARS(6) = 8                     NPLOTIT 3931
CALL GPLOT(XLBL,YLBL,IVARS)      NPLOTIT 3932
IVARS(1) = 3                     NPLOTIT 3933
IVARS(2) = NSTAPS               NPLOTIT 3934
IVARS(3) = 0                     NPLOTIT 3935
120 CALL GPLOT(XP,Y,IVARS)       NPLOTIT 3936
C
C
VARS(1) = 7.                     NPLOTIT 3937
VARS(2) = 8.5                    NPLOTIT 3938
VARS(3) = 0.                     NPLOTIT 3939
VARS(4) = 0.0                    NPLOTIT 3940
VARS(5) = 1.0                    NPLOTIT 3941
VARS(6) = .5                     NPLOTIT 3942
VARS(7) = 1.0                    NPLOTIT 3943
CALL XAXIS(1.2,.5,VARS)          NPLOTIT 3944
CALL CHARS(42,XLBL,0.0,3.5,.05,15) NPLOTIT 3945
VARS(1) = 7.                     NPLOTIT 3946
VARS(2) = 3.8                    NPLOTIT 3947
VARS(3) = 90.                    NPLOTIT 3948
VARS(4) = ATMINP                 NPLOTIT 3949
VARS(5) = ATMAXP                 NPLOTIT 3950
VARS(6) = AINTRV                 NPLOTIT 3951
VARS(7) = 1.                      NPLOTIT 3952
CALL YAXIS(1.2,.5,VARS)          NPLOTIT 3953
CALL CHARS(20,SLEGN,0.0,6.0,4.5,12) NPLOTIT 3954
DO 130 I = 1,NSLICE              NPLOTIT 3955

```

```

JST = NSTAPS*(I-1) NPLOTIT 3961
DO 125 J = 1,NSTAPS NPLOTIT 3962
125 Y(J) = TSM(JST+J) NPLOTIT 3963
SYMBOL = SYMBL(I) NPLOTIT 3964
KS = 0 NPLOTIT 3965
C NPLOTIT 3966
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO NPLOTIT 3967
C IDENTIFY THE CURVES. NPLOTIT 3968
KSTART = I + 1 NPLOTIT 3969
DO 128 K = KSTART,NSTAPS,10 NPLOTIT 3970
KS = KS + 1 NPLOTIT 3971
XLBL(KS) = XS(K) NPLOTIT 3972
128 YLBL(KS) = Y(K) NPLOTIT 3973
IVARS(1) = 6 NPLOTIT 3974
IVARS(2) = NLBLS NPLOTIT 3975
IVARS(3) = 3 NPLOTIT 3976
IVARS(4) = 240+I NPLOTIT 3977
IVARS(5) = 1 NPLOTIT 3978
IVARS(6) = 8 NPLOTIT 3979
CALL GPLOT(XLBL,YLBL,IVARS) NPLOTIT 3980
IVARS(1) = 3 NPLOTIT 3981
IVARS(2) = NSTAPS NPLOTIT 3982
IVARS(3) = 0 NPLOTIT 3983
130 CALL GPLOT(XS,Y,IVARS) NPLOTIT 3984
CALL DISPLA(1) NPLOTIT 3985
C NPLOTIT 3986
C NPLOTIT 3987
150 CONTINUE NPLOTIT 3988
RETURN NPLOTIT 3989
EVD NPLOTIT 3990

```

```

C----SOURCE.NPREPAT NPREPAT 3991
SUBROUTINE PREP(ICHLN,NTTG) NPREPAT 3992
C NPREPAT 3993
C- SOURCE.NPREPAT---- NPREPAT 3994
C NPREPAT 3995
COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000), NPREPAT 3996
Z BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NPREPAT 3997
Z BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80), NPREPAT 3998
Z QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NPREPAT 3999
Z OTIPIN(400), OTIP(80), RHOVG(400), PEX(400), NPREPAT 4000
Z BCTIME(50), TTIO(50), PTIO(50), WPLEN, NPREPAT 4001
Z WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP NPREPAT 4002
C NPREPAT 4003
COMMON /FLMCOL/ RHOVGA(80), PG(80), XPC(80), PLMEFF(80), NPREPAT 4004
Z XMUC(80), EMES(80), REFC(80), NFCSUP(80) NPREPAT 4005
C NPREPAT 4006
COMMON /FRIC/ ALPHA, BETA, DELTA, EPS NPREPAT 4007
C NPREPAT 4008
COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80), NPREPAT 4009
Z CPC(80), GAMC(80), DUMR1(80), DUMR2(80) NPREPAT 4010
C NPREPAT 4011
COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NPREPAT 4012
Z IPLOT, MD1, MD2, MD3, IADJIN, IWRITE NPREPAT 4013
C NPREPAT 4014
COMMON /TCC/ ADUMP, BTA, CD, CP, NPREPAT 4015
Z GAM, PIM, R, SPAN, TOG, NPREPAT 4016
Z WDUMP, WIM, AKC(15,80), AKW(15,80), NPREPAT 4017
Z A(400), AJET(80), AM2(80), CNUM(80), NPREPAT 4018
Z DH(80), DHF(80), DHJ(80), NPREPAT 4019
Z DLX(400), FF(80), HC(80), HG(80), NPREPAT 4020

```

```

Z          P(2,15,80),PEXIT(15),    PUMP(80),    OG(80),      NPREPAT 4021
Z          QSNK(80),   RR(80),      S(15),       T(2,15,400),  NPREPAT 4022
Z          TG(80),     TAU(400),    WPC(80),    XN(80),      NPREPAT 4023
Z          WJ(15,80),   WCROS(2,15,80),  IHUB,        ITIP,       NPREPAT 4024
Z          ICOR,       IFILM,       ISLICE,     NBLKSZ,      NPREPAT 4025
Z          ISBLOK,     NSTA,        NSLICE,     IHC(80)     NPREPAT 4026
Z          NFWD,       NSTA,        SPHTC,      TEPS,       NPREPAT 4027
Z          NFWD,       NSTA,        SPHTM,      TYMMAX     NPREPAT 4028
C          COMMON /TRNSNT/ RHOC,      PHOM,       SPHTC,      SPHTM,      NPREPAT 4029
Z          DLTYME,      TYME,       TEPS,       TYMMAX     NPREPAT 4030
C
C
C          ICHNL IS THE CHANNEL NUMBER; = 1 FOR THE HUB REGION,
C          = NSLICE AT THE TIP
C
C          -- LOCATE INPUT DATA FOR THIS CHANNEL AND STORE IT IN WORKING ARRAYS.
C
C          I1 IS THE STARTING POINT IN THE INDCHN ARRAY FOR THIS CHANNEL
C
C          I1 = INDCHN(ICHL)
C          IF ( ICHNL.NE.INDCHN(I1)) GO TO 290
C
C          -- IF ABOVE TEST IS TRUE, THEN THE DATA IS NOT STORED WHERE EXPECTED
C
10         CONTINUE
          IFILM = INDCHN(I1+1)
          ICOR = INDCHN(I1+2)
          NFWD = INDCHN(I1+3)
          NSTA = INDCHN(I1+4)
          ISPLOK = INDCHN(I1+5)
          NBLKSZ = INDCHN(I1+6)
          IPLOT = INDCHN(I1+7)
          MD1 = INDCHN(I1+8)
          MD2 = INDCHN(I1+9)
          IHUB = INDCHN(I1+12)
          ITIP = INDCHN(I1+13)
          IN1 = I1 + 14
          CD = CHANL(ISBLOK)
          ALPHA = CHANL(ISBLOK+1)
          BETA = CHANL(ISBLOK+2)
          DELTA = CHANL(ISBLOK+3)
          EPS = CHANL(ISBLOK+4)
          ADUMP = CHANL(ISBLOK+6)
          SPAN = CHANL(ISBLOK+7)
          S(ICHL) = SPAN
          BTA = CHANL(ISBLOK+8)
          DLTYME = CHANL(ISBLOK+9)
          TEPS = CHANL(ISBLOK+10)
          NODSF = 5*NFWD
          NODST = 5*NSTA
          I1 = ISBLOK + 14
          I3 = ISBLOK + 14 + 2*NODST
C*****
12         CONTINUE
          DO 205 I = 1,NODST
          IM = I1 + I
          TAU(I) = CHANL(IM)
          A(I) = TAU(I)*SPAN
          NPREPAT 4021
          NPREPAT 4022
          NPREPAT 4023
          NPREPAT 4024
          NPREPAT 4025
          NPREPAT 4026
          NPREPAT 4027
          NPREPAT 4028
          NPREPAT 4029
          NPREPAT 4030
          NPREPAT 4031
          NPREPAT 4032
          NPREPAT 4033
          NPREPAT 4034
          NPREPAT 4035
          NPREPAT 4036
          NPREPAT 4037
          NPREPAT 4038
          NPREPAT 4039
          NPREPAT 4040
          NPREPAT 4041
          NPREPAT 4042
          NPREPAT 4043
          NPREPAT 4044
          NPREPAT 4045
          NPREPAT 4046
          NPREPAT 4047
          NPREPAT 4048
          NPREPAT 4049
          NPREPAT 4050
          NPREPAT 4051
          NPREPAT 4052
          NPREPAT 4053
          NPREPAT 4054
          NPREPAT 4055
          NPREPAT 4056
          NPREPAT 4057
          NPREPAT 4058
          NPREPAT 4059
          NPREPAT 4060
          NPREPAT 4061
          NPREPAT 4062
          NPREPAT 4063
          NPREPAT 4064
          NPREPAT 4065
          NPREPAT 4066
          NPREPAT 4067
          NPREPAT 4068
          NPREPAT 4069
          NPREPAT 4070
          NPREPAT 4071
          NPREPAT 4072
          NPREPAT 4073
          NPREPAT 4074
          NPREPAT 4075
          NPREPAT 4076
          NPREPAT 4077
          NPREPAT 4078
          NPREPAT 4079
          NPREPAT 4080

```

```

IM = IM + NODST          NPREPAT 4081
205 DLX(I) = CHANL(IM)   NPREPAT 4082
DO 215 I = 1,NSTA        NPREPAT 4083
IPLU = 5*I               NPREPAT 4084
DH(I) = 4.0*A(IPLU)/(2.* (SPAN+TAU(IPLU))) NPREPAT 4085
IM = I3 + I               NPREPAT 4086
DHJ(I) = CHANL(IM)       NPREPAT 4087
IM = IM + NSTA           NPREPAT 4088
DHF(I) = CHANL(IM)       NPREPAT 4089
IM = IM + NSTA           NPREPAT 4090
XN(I) = CHANL(IM)        NPREPAT 4091
14  CONTINUE              NPREPAT 4092
IF (DHJ(I).GT.0.0.AND.XN(I).GT.0.0) GO TO 202 NPREPAT 4093
AJET(I) = 0.0             NPREPAT 4094
GO TO 212                NPREPAT 4095
202  CONTINUE              NPREPAT 4096
XOD=XN(I)/DHJ(I)         NPREPAT 4097
IF(XOD.LT.3.1.OR.XOD.GT.12.5) WRITE(6,527) I,XOD NPREPAT 4098
527  FORMAT(1HO,' WARNING, RATIO OF JET HOLE SPACING TO JET DIAMETER
     1'FOR JET ',I2,' IS',F10.4,' WHICH IS OUT OF'
     2' THE RANGE OF VALIDITY OF THE CORRELATION.')
     CNUM(I)=SPAN/XN(I)
C  CNUM IS THE NUMBER OF IMPINGEMENT JETS AT CHANNEL NODE I
C  TOTAL JET AREA IS (AREA OF ONE JET)*(NUMBER OF JETS)
C
54  AJET(I)=3.14159*DHF(I)**2/4.*CNUM(I)          NPREPAT 4103
212  CONTINUE              NPREPAT 4104
     IM = IM + NSTA          NPREPAT 4105
     FR(I) = CHANL(IM)       NPREPAT 4106
     IM = IM + NSTA          NPREPAT 4107
     DP(I) = CHANL(IM)       NPREPAT 4108
     IM = IM + NSTA          NPREPAT 4109
     SP(I) = CHANL(IM)       NPREPAT 4110
     IM = IM + NSTA          NPREPAT 4111
     IM = IM + NSTA          NPREPAT 4112
     INN = INT(I)            NPREPAT 4113
     IHC(I) = INDCHN(INN)    NPREPAT 4114
     IM = IM + NSTA          NPREPAT 4115
     INN = INT(I)            NPREPAT 4116
     IHC(I) = INDCHN(INN)    NPREPAT 4117
215  CONTINUE              NPREPAT 4118
C*****                         NPREPAT 4119
C
C-- NOW, GIVEN SLICE, ICHNL, EVALUATE B.C. AT METAL NODE POINTS. IN THE NPREPAT 4120
C  FOLLOWING:                  NPREPAT 4121
C-- XS & XP ARE DISTANCE FROM LEADING EDGE, ALONG OUTSIDE SURFACE NPREPAT 4122
C  (INCHES), FOR SUCTION & PRESSURE SIDES.                  NPREPAT 4123
C-- THE CONVENTIONS USED IN THE FOLLOWING ARE: INDEX BEGINNING WITH -I- NPREPAT 4124
C-- IS A SLICE INDEX, INDEX BEGINNING WITH -N- IS A TIME INDEX, INDEX NPREPAT 4125
C-- BEGINNING WITH -L- IS A N X INDEX, AND AN INDEX BEGINNING WITH -J- IS NPREPAT 4126
C-- A PROPERTY INDEX I.E. HG,QG,TG,PG.                  NPREPAT 4127
C
C-- FIRST, CHECK THAT THIS IS A TRANSIENT CASE, AND DETERMINE THE MAX. NPREPAT 4128
C-- BCTIME INDEX, NMX.          NPREPAT 4129
     NMX = 1                  NPREPAT 4130
310  IF (BCTIME(NMX+1).LE.0.0) GO TO 315          NPREPAT 4131
     NMX = NMX + 1            NPREPAT 4132
     GO TO 310                NPREPAT 4133
315  CONTINUE              NPREPAT 4134
C
C-- NOW, IF THIS IS A TRANSIENT, FIND THE LOCATION IN THE BCTIME ARRAY NPREPAT 4135
C-- OF THE CURRENT TIME, AND CALCULATE THE VALUE OF THE INTERPOLATING NPREPAT 4136
C-- PARAMETER, TMFRAC.          NPREPAT 4137
                                         NPREPAT 4138
                                         NPREPAT 4139
                                         NPREPAT 4140

```

```

TMFRAC = 0.0 NPREPAT 4141
NLST = 1 NPREPAT 4142
IF (NMX.EQ.1.OR.TYME.LE.0.0) GO TO 330 NPREPAT 4143
C-- THE ABOVE TRANSFER OCCURS IF THIS IS A STEADY STATE PROBLEM NPREPAT 4144
C-- THE FOLLOWING TRANSFER OCCURS IF WE ARE BEYOND THE LAST BCTIME ENTRYNPREPAT 4145
NLST = NMX NPREPAT 4146
IF (TYME.GE.BCTIME(NMX)) GO TO 330 NPREPAT 4147
NMXM1 = NMX - 1 NPREPAT 4148
DO 320 N = 1,NMXM1 NPREPAT 4149
NLST = N NPREPAT 4150
IF (TYME.GE.BCTIME(N).AND.TYME.LT.BCTIME(N+1)) GO TO 325 NPREPAT 4151
320 CONTINUE NPREPAT 4152
325 TMFRAC = (TYME-BCTIME(NLST))/(BCTIME(NLST+1)-BCTIME(NLST)) NPREPAT 4153
C NPREPAT 4154
C-- NEXT, SEARCH THE BCXS & BCXP ARRAYS TO FIND THE X INTERPOLATING NPREPAT 4155
C-- FACTORS, XSP & XPF, FOR POSITIONS XS & XP, SLICE ICHNL. NPREPAT 4156
C-- THE BRACKETING INDICES ARE LBLOWS & LABOVS AND LBLOWP & LABOVP. NPREPAT 4157
C NPREPAT 4158
C-- THE STARTING POINTS IN THE BCXS & BCXP ARRAYS FOR THIS SLICE ARE: NPREPAT 4159
C NPREPAT 4160
330 LSS = (ICHNL-1)*NBCS NPREPAT 4161
LSP = (ICHNL-1)*NBCP NPREPAT 4162
XS = 0.0 NPREPAT 4163
XP = 0.0 NPREPAT 4164
C NPREPAT 4165
C-- THE STARTING POINTS IN THE PROPERTY ARRAYS, FOR THE LATEST TIME STEPNPREPAT 4166
C-- ARE GIVEN BY: NPREPAT 4167
C NPREPAT 4168
JS1S = NSLICE*(NLST-1)*NBCS + LSS NPREPAT 4169
JS1P = NSLICE*(NLST-1)*NBCP + LSP NPREPAT 4170
IF (NMX.EQ.NLST) GO TO 335 NPREPAT 4171
JS2S = JS1S + NSLICE*NBCS NPREPAT 4172
JS2P = JS1P + NSLICE*NBCP NPREPAT 4173
335 HG(1) = BCHGS(JS1S+1) + TMFRAC*(BCHGS(JS2S+1)-BCHGS(JS1S+1)) NPREPAT 4174
TG(1) = BCTGS(JS1S+1) + TMFRAC*(BCTGS(JS2S+1)-BCTGS(JS1S+1)) + 460. NPREPAT 4175
QG(1) = BCQGS(JS1S+1) + TMFRAC*(BCQGS(JS2S+1)-BCQGS(JS1S+1)) NPREPAT 4176
PG(1) = BCPGS(JS1S+1) + TMFRAC*(BCPGS(JS2S+1)-BCPGS(JS1S+1)) NPREPAT 4177
C NPREPAT 4178
DO 350 K = 2,NSTA,2 NPREPAT 4179
C-- THE OUTSIDE SURFACE NODE NUMBERS FOR S & P SIDES ARE: NPREPAT 4180
NNS = 5*K - 4 NPREPAT 4181
NNP = 5*K + 1 NPREPAT 4182
XS = XS + DLX(NNS) NPREPAT 4183
XP = XP + DLX(NNP) NPREPAT 4184
C NPREPAT 4185
DO 340 L = 1,NBCS NPREPAT 4186
LBLOWS = LSS + L - 1 NPREPAT 4187
LABOVS = LSS + L NPREPAT 4188
IF (BCXS(LABOVS).GT.XS) GO TO 342 NPREPAT 4189
340 CONTINUE NPREPAT 4190
C NPREPAT 4191
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXS TABLE NPREPAT 4192
C NPREPAT 4193
342 XSP = (XS-BCXS(LBLOWS))/(BCXS(LABOVS)-BCXS(LBLOWS)) NPREPAT 4194
C NPREPAT 4195
DO 345 L = 1,NBCP NPREPAT 4196
LBLOWP = LSP + L - 1 NPREPAT 4197
LAEOVP = LSP + L NPREPAT 4198
IF (BCXP(LABOVP).GT.XP) GO TO 347 NPREPAT 4199
345 CONTINUE NPREPAT 4200

```

```

C
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXP TABLE
C
347 XPP = (XP-BCXP(LBLOWP))/(BCXP(LABOVP)-BCXP(LBLOWP))
C
C-- NOW THE FRACTIONS ARE KNOWN, CALCULATE THE INTERPOLATED PROPERTIES.
C-- FIRST, FOR THE STEADY STATE OR FOR TIMES BEYOND THE LAST BCTIME:
C
    JB1S = NSLICE*(NLST-1)*NBCS + LBLOWS
    JB1P = NSLICE*(NLST-1)*NBCP + LBLOWP
    HG(K) = BCHGS(JB1S) + XSF*(BCHGS(JB1S+1)-BCHGS(JB1S))
    HG(K+1) = BCHGP(JB1P) + XPP*(BCHGP(JB1P+1)-BCHGP(JB1P))
    QG(K) = BCQGS(JB1S) + XSF*(BCQGS(JB1S+1)-BCQGS(JB1S))
    QG(K+1) = BCQGP(JB1P) + XPP*(BCQGP(JB1P+1)-BCQGP(JB1P))
    TG(K) = BCTGS(JB1S) + XSF*(BCTGS(JB1S+1)-BCTGS(JB1S)) + 460.
    TG(K+1) = BCTGP(JB1P) + XPP*(BCTGP(JB1P+1)-BCTGP(JB1P)) + 460.
    PG(K) = BCPGS(JB1S) + XSF*(BCPGS(JB1S+1)-BCPGS(JB1S))
    PG(K+1) = BCPGP(JB1P) + XPP*(BCPGP(JB1P+1)-BCPGP(JB1P))

C
    IF (NMX.EQ.1.OR.TYME.GE.BCTIME(NMX).OR.TYME.LE.0.0) GO TO 350
    JB2S = NSLICE*NLST*NBCS + LBLOWS
    JB2P = NSLICE*NLST*NBCP + LBLOWP
    AHG = BCHGS(JB2S) + XSF*(BCHGS(JB2S+1)-BCHGS(JB2S))
    HG(K) = HG(K) + TMFRAC*(AHG-HG(K))
    AHG = BCHGP(JB2P) + XPP*(BCHGP(JB2P+1)-BCHGP(JB2P))
    HG(K+1) = HG(K+1) + TMFRAC*(AHG-HG(K+1))
    AQG = BCQGS(JB2S) + XSF*(BCQGS(JB2S+1)-BCQGS(JB2S))
    QG(K) = QG(K) + TMFRAC*(AQG-QG(K))
    AQG = BCQGP(JB2P) + XPP*(BCQGP(JB2P+1)-BCQGP(JB2P))
    QG(K+1) = QG(K+1) + TMFRAC*(AQG-QG(K+1))
    ATG = BCTGS(JB2S) + XSF*(BCTGS(JB2S+1)-BCTGS(JB2S)) + 460.
    TG(K) = TG(K) + TMFRAC*(ATG-TG(K))
    ATG = BCTGP(JB2P) + XPP*(BCTGP(JB2P+1)-BCTGP(JB2P)) + 460.
    TG(K+1) = TG(K+1) + TMFRAC*(ATG-TG(K+1))
    APG = BCPGS(JB2S) + XSF*(BCPGS(JB2S+1)-BCPGS(JB2S))
    PG(K) = PG(K) + TMFRAC*(APG-PG(K))
    APG = BCPGP(JB2P) + XPP*(BCPGP(JB2P+1)-BCPGP(JB2P))
    PG(K+1) = PG(K+1) + TMFRAC*(APG-PG(K+1))

350  CONTINUE
      RETURN
200  WRITE(6,295) ICHNL
205  FORMAT('/* CHANNEL NO. ',I3,' DATA STORAGE IS MESSED UP',
Z     ' *%><;:%%*%&@$***%****')
149  CONTINUE
      RETURN
      END

```

```

C----SOURCE.NTARAYT
      SUBROUTINE TARRAY(JS,JSENS,DELTAN)
C
C- SOURCE.NTARAYT----
C
C++++++ A SUBROUTINE TO SET UP THE COEFFICIENT ARRAY TO SOLVE FOR
C       BLADE TEMPERATURES, TRANSIENT CALCULATIONS.
C       FIRST PUT TOGETHER FROM STEADY STATE PROGRAM, NOVEMBER 24, 1975
C
C*****NTARAYT 4247
C*****NTARAYT 4248
C*****NTARAYT 4249
C*****NTARAYT 4250
C*****NTARAYT 4251
C*****NTARAYT 4252
C*****NTARAYT 4253
C*****NTARAYT 4254
C*****NTARAYT 4255
C*****NTARAYT 4256
C*****NTARAYT 4257
C*****NTARAYT 4258
C*****NTARAYT 4259
C*****NTARAYT 4260

```

C NODE NAMES FOR STATION IS ARE:
 C LCOOL = COOLANT NODE AT IS.
 C LIN = INSIDE WALL NCDE.
 C L = MID WALL NODE.
 C LOUT = OUTSIDE WALL NODE.
 C LCUP = ADJACENT UPSTREAM COOLANT NODE.
 C LCUPS = ADJACENT INSIDE WALL NODE
 C LUP = ADJACENT UPSTREAM MID-WALL NODE.
 C LDN = ADJACENT DOWNSTREAM MID-WALL NODE.
 C LJ = JUNCTION OF COATING AND WALL METAL
 C
 C IS
 C |
 C -----LOUT-*-----
 C -----LJ *-----
 C ////////////////
 C WALL //LUP-*//L-*//LDN-*//
 C ////////////////
 C -LCUPS-*---LIN-*---
 C FLOW
 C ---> LCUP-* LCOOL-*
 C
 C -LCUPP-*--LCOOLP-*-----
 C
 C PRESSURE SIDE WALL OR PLENUM
 C
 C*****NTARAYT 4290
 C*****NTARAYT 4291
 C NTARAYT 4292
 C NTARAYT 4293
 C REAL*8 TCOF
 COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000), NTARAYT 4294
 Z BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NTARAYT 4295
 Z BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80), NTARAYT 4296
 Z QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NTARAYT 4297
 Z QTIPIN(400), QTIP(80), RHOVG(400), PEX(400), NTARAYT 4298
 Z BCTIME(50), TTIO(50), PTIO(50), WPLEN, NTARAYT 4299
 Z WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP NTARAYT 4300
 C
 COMMON /PLMCOL/ RHOVGA(80), PG(80), XFC(80), FLMEFF(80), NTARAYT 4301
 Z XMUC(80), EMES(80), REFC(80), NFCSUP(80) NTARAYT 4302
 C
 COMMON /MATRX/ TCOP(400,30)
 C
 COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80), NTARAYT 4305
 Z CPC(80), GAMC(80), DUMR1(80), DUMR2(80) NTARAYT 4306
 C
 COMMON /TCO/ ADUMP, BTA, CD, CP, NTARAYT 4310
 Z GAM, PIM, R, SPAN, TOG, NTARAYT 4311
 Z WDUMP, WIM, AKC(15,80), AKW(15,80), NTARAYT 4312
 Z A(400), AJET(80), AM2(80), CNUM(80), NTARAYT 4313
 Z DH(80), DHF(80), DHJ(80), NTARAYT 4314
 Z DLX(400), FF(80), HC(80), HG(80), NTARAYT 4315
 Z P(2,15,80), PEXIT(15), PUMP(80), QG(80), NTARAYT 4316
 Z QSNK(80), RR(80), S(15), T(2,15,400), NTARAYT 4317
 Z TG(80), TAU(400), WFC(80), NTARAYT 4318
 Z WJ(15,80), WCROS(2,15,80), XN(80), NTARAYT 4319
 Z ICOR, IFILM, IHUB, ITIF, NTARAYT 4320

```

Z           ISBLOK,      ISLICE,      NBLKSZ,      NSLICE,
Z           NFWD,        NSTA,        IHC(80)          NTARAYT 4321
C           COMMON /TRNSNT/ RHOC,      RHOM,        SPHTC,      SPHTM,
Z           DLTYME,      TYME,        TEPS,        TYMMAX          NTARAYT 4322
C           DIMENSION EPAREA(80), DELTAN(15)          NTARAYT 4323
C           RTRNVM = 0.
C           RCHRDW = 0.
C           TREPS = 1.0
C           IF (TYME.GE.0.) TREPS = TEPS          NTARAYT 4324
C           HX = 1.0
C           RCVRY = .89          NTARAYT 4325
300         CONTINUE          NTARAYT 4326
C           SPAN = S(ISLICE)          NTARAYT 4327
C           NODST = 5*NSTA          NTARAYT 4328
C           NODSF = 5*NFWD          NTARAYT 4329
C           DO 308 J = 1,30          NTARAYT 4330
C           DO 308 I = 1,400          NTARAYT 4331
308         TCOF(I,J) = 0.0          NTARAYT 4332
C           DO 309 I = 1,9          NTARAYT 4333
C           ICO = NODST + I          NTARAYT 4334
309         DLX(ICO) = 0.0          NTARAYT 4335
C           ICOMS = JS + 2 - 4*JSENS          NTARAYT 4336
C           ICOMP = JS - 2 + 4*JSENS          NTARAYT 4337
C**** ICOMS IS STATION ADJACENT TO JS, IN SUCTION DIRECTION          NTARAYT 4338
C**** ICOMP IS STATION ADJACENT TO JS, IN PRESSURE DIRECTION          NTARAYT 4339
C           IP (ICOMS.LT.0) ICOMS=2          NTARAYT 4340
C           IP (ICOMP.LT.1) ICOMP=1          NTARAYT 4341
310         CONTINUE          NTARAYT 4342
C           NTARAYT 4343
C           NTARAYT 4344
C           NTARAYT 4345
C           BEGIN OVERALL LOOP, WHERE LOOP VARIABLE (IS) IS THE STATION NUMBER          NTARAYT 4346
C
C           DO 440 IS = 1,NSTA          NTARAYT 4347
C           ISUP = IS - 2          NTARAYT 4348
C           YIMP = 0.0          NTARAYT 4349
C           YFINS = 0.0          NTARAYT 4350
C           YCONV = 0.0          NTARAYT 4351
C           YIMPU = 0.0          NTARAYT 4352
C           YFINSU = 0.0          NTARAYT 4353
C           YCONVU = 0.0          NTARAYT 4354
C           YIMPUU = 0.0          NTARAYT 4355
C           YPNSUU = 0.0          NTARAYT 4356
C           YCNVUU = 0.0          NTARAYT 4357
C           IF (IHC(IS).EQ.1) YIMP=1.0          NTARAYT 4358
C           IF (IHC(IS).EQ.2) YCONV= (1.0+RCVRY*AM2(IS)*(GAMC(IS)-1.)/2.)          NTARAYT 4359
C           IF (IHC(IS).EQ.3) YFINS=1.0          NTARAYT 4360
C           FACTOR = 1.0          NTARAYT 4361
C           IF (IS.EQ.ICOMS.OR.IS.EQ.ICOMP) FACTOR = .5          NTARAYT 4362
C           ISENS = 0          NTARAYT 4363
C           ISEN = IS - 2*(IS/2)          NTARAYT 4364
C           NTARAYT 4365
C           NTARAYT 4366
C           NTARAYT 4367
C           NTARAYT 4368
C           NTARAYT 4369
C           NTARAYT 4370
C           NTARAYT 4371
C           NTARAYT 4372
C           IF (IS.GT.NFWD+1)-- IN TRAILING EDGE REGION, GO TO 380          NTARAYT 4373
C
C           IP(IS.GT.NFWD+1) GO TO 380          NTARAYT 4374
C
C           J9 = 16          NTARAYT 4375
C           LCOOL = 5*IS          NTARAYT 4376
C           LIN = LCOOL-1          NTARAYT 4377
C           L = LCOOL-2          NTARAYT 4378
C           NTARAYT 4379
C           NTARAYT 4380

```

```

LJ      =LCOOL - 3          NTARAYT 4381
LOUT    =LCOOL-4          NTARAYT 4382
IF (IHC(IS).NE.2) GO TO 320  NTARAYT 4383
C
C   FORCED CONVECTION HC:
TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.  NTARAYT 4384
CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)  NTARAYT 4385
RE(IS) = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH (IS)/(A(LCOOL)*XMU)  NTARAYT 4386
HC (IS) = .023*12.* (C/DH (IS))*(RE (IS)**.8)*(PD**.333)  NTARAYT 4387
NTARAYT 4388
NTARAYT 4389
NTARAYT 4390
NTARAYT 4391
NTARAYT 4392
NTARAYT 4393
NTARAYT 4394
NTARAYT 4395
NTARAYT 4396
NTARAYT 4397
NTARAYT 4398
NTARAYT 4399
NTARAYT 4400
NTARAYT 4401
NTARAYT 4402
NTARAYT 4403
NTARAYT 4404
NTARAYT 4405
NTARAYT 4406
NTARAYT 4407
NTARAYT 4408
NTARAYT 4409
NTARAYT 4410
NTARAYT 4411
NTARAYT 4412
NTARAYT 4413
NTARAYT 4414
NTARAYT 4415
NTARAYT 4416
NTARAYT 4417
NTARAYT 4418
NTARAYT 4419
NTARAYT 4420
NTARAYT 4421
NTARAYT 4422
NTARAYT 4423
NTARAYT 4424
NTARAYT 4425
NTARAYT 4426
NTARAYT 4427
NTARAYT 4428
NTARAYT 4429
NTARAYT 4430
NTARAYT 4431
NTARAYT 4432
NTARAYT 4433
NTARAYT 4434
NTARAYT 4435
NTARAYT 4436
NTARAYT 4437
NTARAYT 4438
NTARAYT 4439
NTARAYT 4440
C
320  CONTINUE
IF (IS.GE.JS) GO TO 322
C
C   SPECIAL CASE FOR STATION NUMBER 1:
IF (IS.EQ.1) ISUP = 2
C
C   IF STATION IS IS FORWARD OF FLOW SPLIT, AND ON SAME SIDE, GO TO 370
IF (ISEN.EQ.JSENS) GO TO 370
C
322  CONTINUE
IF (ISUP.LT.1) ISUP = 1
IF (IHC(ISUP).EQ.1) YIMPU = 1.0
IF (IHC(ISUP).EQ.2) YCONVU=1.0+RCVRY*AM2(ISUP)*(GAMC(ISUP)-1.)/2.
IF (IHC(ISUP).EQ.3) YFINSU = 1.0
LCUP = LCOOL - 10
LCUPS = LCOOL - 11
LUP = LCOOL - 12
LDN = LCOOL + 8
IF (IS.NE.2) GO TO 324
C
C   IF THIS IS STATION NUMBER 2:
LCUP = 5
LCUPS = 4
LUP = 3
324  CONTINUE
IF (IS.GT.1) GO TO 326
C
C   IF THIS IS STATION NUMBER 1:
LCUP = 10
LCUPS = 9
LUP = 8
326  CONTINUE
C
C   IS = 1, STATION NO. 1, LEADING EDGE NODES
C
IF (IS.NE.JS) GO TO 330
C
*****
***** THIS BLOCK COMPUTES TCOF ELEMENTS FOR THE STATION AT WHICH  NTARAYT 4430
***** THE FLOW SPLITS  NTARAYT 4431
***** IS = JS  NTARAYT 4432
*****
*****  NTARAYT 4433
*****  NTARAYT 4434
*****
328  CONTINUE
DX1 = DLX(LUP)
IF (DX1.EQ.0.0) DX1 = DLX(L)
DX2 = DLX(LDN)
IF (DX2.EQ.0.0) DX2 = DLX(L)

```

```

DX3 = DLX(LUP-2)
IF (DX3.EQ.0.0) DX3 = DLX(LOUT)
DX4 = DLX(LDN-2)
IF (DX4.EQ.0.0) DX4 = DLX(LOUT)
DX5 = DLX(LUP+1)
IF (DX5.EQ.0.0) DX5 = DLX(LIN)
DX6 = DLX(LDN+1)
IF (DX6.EQ.0.0) DX6 = DLX(LIN)
DX9 = DLX(LUP-1)
IF (DX9.EQ.0.0) DX9 = DLX(LJ)
DX10 = DLX(LDN-1)
IF (DX10.EQ.0.) DX10 = DLX(LJ)
CURV = 1.0 + (DX9+DX10)/(DX3+DX4)

C CURV IS A MEASURE OF THE CURVATURE OF THE BLADE AT STATION IS.
C CURV = 2.0 IS A STRAIGHT SECTION OF WALL,
C CURV < 2.0 IS A CONVEX SECTION, AND
C CURV > 2.0 IS A CONCAVE SECTION

C TRTRMC = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMC=(3600./144.)*RHOC*SPHTC*
Z (TAU(LOUT)**2)/(4.*AKC(ISLICE,IS)*DLTYME)
TCOF(LOUT,13) = -TREPS + TRTRMC*CURV
TCOF(LOUT,12)=TRTRMC*CURV + TREPS*(1.0+(1.0-BTA)*HG(IS)*TAU(LOUT)/
Z (12.*AKC(ISLICE,IS))) NTARAYT 4455
TCOF(LOUT,J9) = -(1.-BTA)*TREPS*FLMEFF(IS)*HG(IS)*TAU(LOUT)/
Z (12.*AKC(ISLICE,IS)) NTARAYT 4456
TCOF(LOUT,24) = (BTA*QG(IS) + (1.0-BTA)*HG(IS)*TG(IS)*
Z (1.0-FLMEFF(IS)))*TAU(LOUT)/(12.*AKC(ISLICE,IS))
Z - T(1,ISLICE,LOUT)*((1.-TREPS)*((1.-BTA)*HG(IS)*TAU(LOUT)/
Z (12.*AKC(ISLICE,IS)) + 1.) - TRTRMC*CURV)
Z + T(1,ISLICE,LJ)*(1.-TREPS+TRTRMC*CURV)
Z + T(1,ISLICE,LCOOL)*(1.-TREPS)*FLMEFF(IS)*(1.-BTA)*HG(IS)*
Z TAU(LOUT)/(12.*AKC(ISLICE,IS))

C TCOF(LJ,11) = TREPS
TCOF(LJ,13) = TREPS*(AKW(ISLICE,IS)/AKC(ISLICE,IS))*(
Z (2.*TAU(LOUT)/TAU(L))*(DX1+DX2+DX3+DX4)/(DX9+DX10+DX3+DX4)) NTARAYT 4460
TCOF(LJ,12) = - TCOF(LJ,11) - TCOF(LJ,13)
TCOF(LJ,24) = (1.-TREPS)*((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) +
Z (T(1,ISLICE,LJ)-T(1,ISLICE,L))*TCOF(LJ,13)/TREPS) NTARAYT 4461

C J1 = 12 - L + LUP
J2 = 12 - L + LDN

C THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10) NTARAYT 4462
THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10) NTARAYT 4463
THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))*2.*TAU(L)/(DX1+DX2+DX9+DX10) NTARAYT 4464
THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10)) NTARAYT 4465
THETA4 = 0.0 NTARAYT 4466
THETA5 = 0.0 NTARAYT 4467
HUB1 = 0.0 NTARAYT 4468
HUB3 = 0.0 NTARAYT 4469
TIP1 = 0.0 NTARAYT 4470
TIP3 = 0.0 NTARAYT 4471
IF (ISLICE.EQ.1) GO TO 3290 NTARAYT 4472

C FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE:
C THETA4 = (TAU(L)/S(ISLICE))*(TAU(L)/(S(ISLICE)+S(ISLICE-1)))* NTARAYT 4473
NTARAYT 4474
NTARAYT 4475
NTARAYT 4476
NTARAYT 4477
NTARAYT 4478
NTARAYT 4479
NTARAYT 4480
NTARAYT 4481
NTARAYT 4482
NTARAYT 4483
NTARAYT 4484
NTARAYT 4485
NTARAYT 4486
NTARAYT 4487
NTARAYT 4488
NTARAYT 4489
NTARAYT 4490
NTARAYT 4491
NTARAYT 4492
NTARAYT 4493
NTARAYT 4494
NTARAYT 4495
NTARAYT 4496
NTARAYT 4497
NTARAYT 4498
NTARAYT 4499
NTARAYT 4500

```

```

Z           2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)          NTARAYT 4501
IF (ISLICE.EQ.NSLICE) GO TO 3292          NTARAYT 4502
THETA5 = THETA4*(S(ISLICE)+S(ISLICE-1))/(S(ISLICE)+S(ISLICE+1))  NTARAYT 4503
TBELOW = T(1,ISLICE-1,L)          NTARAYT 4504
TABOVE = T(1,ISLICE+1,L)          NTARAYT 4505
GO TO 3294          NTARAYT 4506
C
C FOR THE SLICE AT THE HUB END OF THE BLADE:          NTARAYT 4507
C
3290 CONTINUE          NTARAYT 4508
IF (IHUB.EQ.1) HUB1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*          NTARAYT 4509
Z           (2.*(TAU(L)/S(1))**2)          NTARAYT 4510
C FOR IHUB = 1, HUB TEMPERATURE IS SPECIFIED*****          NTARAYT 4511
THETA5 = 0.0          NTARAYT 4512
IF (NSLICE.GT.1) THETA5 = (TAU(L)/S(1))*(TAU(L)/(S(1)+S(2)))*          NTARAYT 4513
Z           (2.*(DX1+DX2)/(DX1+DX2+DX9+DX10))          NTARAYT 4514
IF (IHUB.EQ.3) HUB3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/          NTARAYT 4515
Z           (AKW(1,IS)*12.*S(1))          NTARAYT 4516
C IHUB = 3, THE HEAT FLUX AT THE HUB END IS SPECIFIED (BTU/HR PT**2) ***          NTARAYT 4517
TBELOW = T(1,1,L)          NTARAYT 4518
IF (IHUB.EQ.1) TBELOW = THUB(IS)          NTARAYT 4519
TABOVE = T(1,1,L)          NTARAYT 4520
IF (NSLICE.GT.1) TABOVE = T(1,2,L)          NTARAYT 4521
IF (NSLICE.GT.1) GO TO 3294          NTARAYT 4522
C
C FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES          NTARAYT 4523
BEING CONSIDERED) : *****          NTARAYT 4524
C
3292 CONTINUE          NTARAYT 4525
IF (ITIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*          NTARAYT 4526
Z           (2.*(TAU(L)/S(NSLICE))**2)          NTARAYT 4527
IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L)          NTARAYT 4528
IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/          NTARAYT 4529
Z           (AKW(NSLICE,IS)*12.*S(NSLICE))          NTARAYT 4530
TABOVE = T(1,ISLICE,L)          NTARAYT 4531
IF (ITIP.EQ.1) TABOVE = TTIP(IS)          NTARAYT 4532
C
C
3294 CONTINUE          NTARAYT 4533
THETA9 = 0.0          NTARAYT 4534
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) THETA9 = 2.*3600.*RHOM*SPHTM*          NTARAYT 4535
Z           (DX1+DX2)*(TAU(L)**2)/(144.*AKW(ISLICE,IS)*          NTARAYT 4536
Z           (DX1+DX2+DX9+DX10)*DLTYME)          NTARAYT 4537
C
TCOF(L,11) = 1.0*TREPS          NTARAYT 4538
TCOF(L,13) = THETA1*TREPS          NTARAYT 4539
TCOF(L,J1) = THETA2*TREPS          NTARAYT 4540
TCOF(L,J2) = THETA3*TREPS          NTARAYT 4541
TCOF(L,12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5 -          NTARAYT 4542
Z           HUB1 - TIP1)*TREPS - THETA9          NTARAYT 4543
TCOF(L,24) = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW          NTARAYT 4544
Z           -(THETA5+TIP1)*TABOVE - QHUB(IS)*HUB3 + QTIP(IS)*TIP3          NTARAYT 4545
Z           -(1-TREPS)*(T(1,ISLICE,LUP)*THETA2+T(1,ISLICE,LJ)          NTARAYT 4546
Z           +T(1,ISLICE,LIN)*THETA1+T(1,ISLICE,LDN)*THETA3)          NTARAYT 4547
Z           +T(1,ISLICE,L)*(1.0-TREPS)*(1.+THETA1+THETA2+THETA3          NTARAYT 4548
Z           +THETA4+THETA5+HUB1+TIP1) - THETA9)          NTARAYT 4549
C
C
AHTRN1 = (DX5 + DX6)*S(ISLICE)/2.          NTARAYT 4550
THETA8 = 2.*HX*HC(IS)*AHTRN1*TAU(L)/(12.*AKW(ISLICE,IS)*S(ISLICE))          NTARAYT 4551

```



```

AHTRN1 = (DX5 + DX6)*S(ISLICE)/2.          NTARAYT 4621
A1 = SPAN*DX5/2.                           NTARAYT 4622
A2 = A1                                     NTARAYT 4623
A3 = 0.0                                     NTARAYT 4624
A4 = 0.0                                     NTARAYT 4625
C
C      IF (IHC (IS) .EQ. 3.AND.IS.LE.NPWD)
Z      CALL HCPINS (IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)
IF (IHC (IS-2) .EQ. 3) A1      = EFAREA (IS-2)/2.
IF (IHC (IS) .EQ. 3) A2      = EFAREA (IS)/2.
IF (IHC (IS) .EQ. 3) AHTRN1 = EFAREA (IS)
C
C      I3 = 12 - LIN + LCOOL
340  CONTINUE
C
C      CURV = 1.0 + (DX9+DX10)/(DX3+DX4)
C
C***** *****
C FOR MID-METAL NODE (L) :           IN GENERAL; I3 = 13 NTARAYT 4640
C      TCOF(L,12), 12 REFERS TO NODE L
C      (L,J1), J1 REFERS TO NODE LUP   J1 = 2 NTARAYT 4641
C      (L,J2), J2 REFERS TO NODE LDN   J2 = 22 NTARAYT 4642
C      (L,11), 11 REFERS TO NODE LJ    J4 = 1 NTARAYT 4643
C      (L,13), 13 REFERS TO NODE LIN   J5 = 2 NTARAYT 4644
C
C FOR THE COOLANT NODE (LCOOL) :       J6 = 6 NTARAYT 4645
C      TCOF(LCOOL,12), 12 REFERS TO NODE LCOOL   J8 = 16 NTARAYT 4646
C      (LCOOL,J4), J4 REFERS TO NODE LCUPS = LUP+1   NTARAYT 4647
C      ,J5), J5 REFERS TO NODE LCUP   NTARAYT 4648
C      ,J6), J6 REFERS TO NODE LCUPP, (TRAILING EDGE REGION ONLY) NTARAYT 4649
C      ,11), 11 REFERS TO NODE LIN   NTARAYT 4650
C      ,J8), J8 REFERS TO NODE LCOOLP, (TRAILING EDGE REGION ONLY) NTARAYT 4651
C
C***** *****
C FOR OUTSIDE NODE:
C
C      TRTRMC = 0.0
C      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMC =
Z      (3600./144.)*RHOC*SPHTC*(TAU(LOUT)**2)/(4.*AKC (ISLICE,IS) *DLTYME) NTARAYT 4652
TCOF(LOUT,13) = -TREPS + TRTRMC*CURV NTARAYT 4653
TCOF(LOUT,12) = (1.0 + (1.0-BTA)*HG (IS)*TAU(LOUT)/
Z                  (12.*AKC (ISLICE,IS))) *TREPS + TRTRMC*CURV NTARAYT 4654
TCOF(LOUT,J9) = -(1.-BTA)*TREPS*FLMEFF (IS)*HG (IS)*TAU(LOUT)/
Z                  (12.*AKC (ISLICE,IS))) NTARAYT 4655
TCOF(LOUT,24) = (BTA*QG (IS) + (1.0-BTA)*HG (IS)*TG (IS)*
Z                  (1.0-FLMEFF (IS)))*TAU(LOUT)/(12.*AKC (ISLICE,IS)) NTARAYT 4656
Z - T(1,ISLICE,LOUT)*((1.-TREPS)*((1.-BTA)*HG (IS)*TAU(LOUT)/
Z                  (12.*AKC (ISLICE,IS)) + 1.) - TRTRMC*CURV) NTARAYT 4657
Z + T(1,ISLICE,LJ)*(1.-TREPS+TRTRMC*CURV) NTARAYT 4658
Z + T(1,ISLICE,LCOOL)*(1.-TREPS)*FLMEFF (IS)*(1.-BTA)*HG (IS)*
Z      TAU(LOUT)/(12.*AKC (ISLICE,IS)) NTARAYT 4659
C
C AT JUNCTION OF COATING AND METAL, NODE LJ:
C
TCOF(LJ,11) = TREPS NTARAYT 4660
TCOF(LJ,13) = TREPS*(AKW (ISLICE,IS)/AKC (ISLICE,IS))* NTARAYT 4661
Z      (2.*TAU (LOUT)/TAU (L))* (DX1+DX2+DX3+DX4)/(DX9+DX10+DX3+DX4) NTARAYT 4662
TCOF(LJ,12) = - TCOF(LJ,11) - TCOF(LJ,13) NTARAYT 4663
NTARAYT 4664
NTARAYT 4665
NTARAYT 4666
NTARAYT 4667
NTARAYT 4668
NTARAYT 4669
NTARAYT 4670
NTARAYT 4671
NTARAYT 4672
NTARAYT 4673
NTARAYT 4674
NTARAYT 4675
NTARAYT 4676
NTARAYT 4677
NTARAYT 4678
NTARAYT 4679
NTARAYT 4680

```

```

TCOP(LJ,24) = (1.-TREPS)*((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) +
Z      (T(1,ISLICE,LJ)-T(1,ISLICE,L))*TCOP(LJ,13)/TREPS)          NTARAYT 4681
C
C FOR MID-METAL NODE:
C
THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10)          NTARAYT 4682
THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)  NTARAYT 4683
THETA3 = 0.0                                              NTARAYT 4684
THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10))  NTARAYT 4685
IF (IS.LT.NSTA-1) THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))*2.*        NTARAYT 4686
Z      TAU(L)/(DX1+DX2+DX9+DX10)                           NTARAYT 4687
THETA4 = 0.0                                              NTARAYT 4688
THETA5 = 0.0                                              NTARAYT 4689
HUB1 = 0.0                                                NTARAYT 4690
HUB3 = 0.0                                                NTARAYT 4691
TIP1 = 0.0                                                NTARAYT 4692
TIP3 = 0.0                                                NTARAYT 4693
IF (ISLICE.EQ.1) GO TO 3410                            NTARAYT 4694
C
C FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE:
C
THETA4 = (TAU(L)/S(ISLICE))*(TAU(L)/(S(ISLICE)+S(ISLICE-1)))*    NTARAYT 4695
Z      2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)                      NTARAYT 4696
IF (ISLICE.EQ.NSLICE) GO TO 3412                        NTARAYT 4697
THETA5 = THETA4*(S(ISLICE)+S(ISLICE-1))/(S(ISLICE)+S(ISLICE+1))  NTARAYT 4698
TBELOW = T(1,ISLICE-1,L)                                NTARAYT 4699
TABOVE = T(1,ISLICE+1,L)                                NTARAYT 4700
GO TO 3414                                              NTARAYT 4701
C
C FOR THE SLICE AT THE HUB END OF THE BLADE:
C
3410 CONTINUE
IF (IHUB.EQ.1) HUB1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(2.*(TAU(L)/
Z      S(1))**2)                                         NTARAYT 4712
C FOR IHUB = 1, HUB TEMPERATURE IS SPECIFIED*****          NTARAYT 4713
C
THETA5 = 0.0                                              NTARAYT 4714
IF (NSLICE.GT.1) THETA5 = (TAU(L)/S(1))*(TAU(L)/(S(1)+S(2)))*
Z      (2.*(DX1+DX2)/(DX1+DX2+DX9+DX10))                NTARAYT 4715
C
IF (IHUB.EQ.3) HUB3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/
Z      (AKW(1,IS)*12.*S(1))                                NTARAYT 4716
C IHUB = 3, THE HEAT FLUX AT THE HUB END IS SPECIFIED (BTU/HR FT**2) ***NTARAYT 4717
C
TBELOW = T(1,1,L)                                         NTARAYT 4718
IF (IHUB.EQ.1) TBELOW = THUB(IS)                         NTARAYT 4719
C
TABOVE = T(1,1,L)                                         NTARAYT 4720
IF (NSLICE.GT.1) TABOVE = T(1,2,L)                         NTARAYT 4721
C
IF (NSLICE.GT.1) GO TO 3414                            NTARAYT 4722
C
C FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES
C BEING CONSIDERED) : *****
C
3412 CONTINUE
IF (ITIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(2.*(TAU(L)/
Z      S(NSLICE))**2)                                     NTARAYT 4723
IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L)                  NTARAYT 4724

```

```

IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/ NTARAYT 4741
Z (AKW(NSLICE,IS)*12.*S(NSLICE)) NTARAYT 4742
T ABOVE = T(1,ISLICE,L) NTARAYT 4743
IF (ITIP.EQ.1) TABOVE = TTIP(IS) NTARAYT 4744
C NTARAYT 4745
3414 CONTINUE NTARAYT 4746
THETA9 = 0.0 NTARAYT 4747
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) THETA9 = 2.*3600.*RHOM*SPHTM* NTARAYT 4748
Z (DX1+DX2)*(TAU(L)**2)/(144.*AKW(ISLICE,IS)* NTARAYT 4749
Z (DX1+DX2+DX9+DX10)*DLTYME) NTARAYT 4750
C NTARAYT 4751
C ENDEFF IS EFFECT OF HEAT TRANSFER FROM THE GAS TO THE REAR EDGE OF NTARAYT 4752
C THE BLADE NTARAYT 4753
C ENDEFF = 0.0 NTARAYT 4754
ENDFLX = 0.0 NTARAYT 4755
IF (IS.GE.NSTA-1.AND.BTA.EQ.0.0) ENDEFF = 2.*HG(IS)*(TAU(L)**2)/ NTARAYT 4756
Z (12.*AKW(ISLICE,IS)*(DX1+DX9)) NTARAYT 4757
IF (IS.GE.NSTA-1.AND.BTA.GT.0.0) ENDFLX = QG(IS)*(TAU(L)**2)/ NTARAYT 4758
Z (12.*AKW(ISLICE,IS)*(DX1+DX9)) NTARAYT 4759
3416 CONTINUE NTARAYT 4760
TCOP(L,11) = 1.0*TREPS NTARAYT 4761
TCOP(L,13) = THETA1*TREPS NTARAYT 4762
TCOP(L,J1) = THETA2*TREPS NTARAYT 4763
TCOP(L,J2) = 0.0 NTARAYT 4764
TCOP(L,12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5 NTARAYT 4765
- ENDEFF - HUB1 - TIP1)*TREPS - THETA9 NTARAYT 4766
TCOP(L,24) = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW - NTARAYT 4767
Z (THETA5+TIP1)*T ABOVE - QHUB(IS)*HUB3 + QTIP(IS)*TIP3 NTARAYT 4768
Z -(1.-TREPS)*(T(1,ISLICE,LUP)*THETA2+T(1,ISLICE,LJ)+ NTARAYT 4769
Z T(1,ISLICE,LIN)*THETA1+T(1,ISLICE,LDN)*THETA3) NTARAYT 4770
Z +T(1,ISLICE,L)*((1.0-TREPS)*(1.+THETA1+THETA2+THETA3+ NTARAYT 4771
Z THETA4+THETA5+HUB1+TIP1+ENDEFF) NTARAYT 4772
Z - THETA9) - TG(IS)*ENDEFF - ENDFLX NTARAYT 4773
Z IF (IS.LT.NSTA-1) TCOP(L,J2) = THETA3*TREPS NTARAYT 4774
C NTARAYT 4775
PUMP(IS) = (.1047198*WS)**2*RR(IS)*(RR(IS)-RR(IS-2)) NTARAYT 4776
C NTARAYT 4777
C FOR INNER SURFACE NODE:
C
342 THETA8 = 2.*HX*HC(IS)*AHTRN1*TAU(L)/(12.*AKW(ISLICE,IS)* NTARAYT 4778
Z S(ISLICE)*(DX1+DX2+DX5+DX6)) NTARAYT 4779
C
TCOP(LIN,11) = TREPS NTARAYT 4780
TCOP(LIN,12) = (-1.0 - THETA8)*TREPS NTARAYT 4781
TCOP(LIN,I3) = THETA8*(YCONV + YFINS)*TREPS NTARAYT 4782
TCOP(LIN,24) = - YIMP*THETA8*T0G NTARAYT 4783
Z -(1.-TREPS)*(T(1,ISLICE,L)-T(1,ISLICE,LIN)*(1.+THETA8)) NTARAYT 4784
Z +THETA8*(YCONV+YFINS)*T(1,ISLICE,LCOOL)) NTARAYT 4785
C
C IF THIS IS A TRAILING EDGE, PRESSURE SIDE, STATION, COOLANT NODE NTARAYT 4786
C COINCIDES WITH SUCTION SIDE COOLANT NODE. NTARAYT 4787
IF (ISENS.EQ.0) GO TO 343 NTARAYT 4788
TCOP(LCOOL,7) = -1.0 NTARAYT 4789
TCOP(LCOOL,12) = 1.0 NTARAYT 4790
TCOP(LCOOL,24) = 0.0 NTARAYT 4791
IF (ISENS.EQ.1) GO TO 430 NTARAYT 4792
343 CONTINUE NTARAYT 4793
C NTARAYT 4794

```

```

C FOR COOLANT NODE:
C FOR THE SPECIAL CASE OF IS = NFWD+1, GO TO 350
C
C FILMW = WFC(IS)
IF (IS.GT.NFWD) FILMW = FILMW + WFC(IS+1)
IF (IS.EQ.NFWD+1) GO TO 350
C
C
WXCP = WCROS(2,ISLICE,IS)*144.*CPC(IS)*3600.
IF (IS.EQ.ICOMP.OR.IS.EQ.ICOMS) WXCP = WJ(ISLICE,JS)*144.*CPC(IS)*3600./2.
Z
C
C DEFINE A COOLANT SIDE TRANSIENT TERM, TRTRMG
C
TRTRMG = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMG = (1.+CPC(IS-2)/CPC(IS))*NTARAYT 4801
Z (P(1,ISLICE,IS-2)/T(1,ISLICE,LCUP)) NTARAYT 4802
Z - P(1,ISLICE,IS)/T(1,ISLICE,LCOOL)) NTARAYT 4803
Z *(A(LCUP)+A(LCOOL))/DLX(LCOOL)/(16.*DLTYME*F*) NTARAYT 4804
Z WCROS(2,ISLICE,IS)*12.) NTARAYT 4805
TCOF(LCOOL,J4) = HX*TREPS*HC(ISUP)*A1/WXCP NTARAYT 4806
TCOF(LCOOL,J5) = 144.*3600.*TREPS*WCROS(2,ISLICE,ISUP)*NTARAYT 4807
Z (1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)*CPC(ISUP)/WXCP NTARAYT 4808
Z - HX*TREPS*HC(ISUP)*(YCONVU+YPINSU)*(A1+A3)/WXCP - TRTRMG NTARAYT 4809
TCOF(LCOOL,11) = HX*TREPS*HC(IS)*A2/WXCP NTARAYT 4810
TCOF(LCOOL,12) = TREPS*((-FILMW*144.*3600.*CPC(IS)/WXCP)-1.0 NTARAYT 4811
Z -(GAMC(IS)-1.)/2.)*AM2(IS) NTARAYT 4812
Z - HX*HC(IS)*(A2+A4)*(YCONV+YPINS)/WXCP - TRTRMG NTARAYT 4813
TREDGE = 0.0 NTARAYT 4814
IF (IS.GT.NFWD) TREDGE=((1.-TREPS)*HX/WXCP)*(T(1,ISLICE,LCUPP)*NTARAYT 4815
Z HC(ISUP)*A3 + T(1,ISLICE,LCOOLP)*HC(IS)*A4) NTARAYT 4816
TCOF(LCOOL,24) = -(CPO*FACTOR*WJ(ISLICE,ISUP)*144.*3600./WXCP)*TOGNNTARAYT 4817
Z - PUMP(IS)/(CPC(IS)*778.*144.*32.2) NTARAYT 4818
Z + TOG*(HX*HC(ISUP)*A1*YIMPU + HX*HC(IS)*A2*YIMP)/WXCP NTARAYT 4819
Z - T(1,ISLICE,LCUPS)*HX*(1-TREPS)*HC(ISUP)*A1/WXCP NTARAYT 4820
Z - T(1,ISLICE,LCUP)*(144.*3600.*(1.-TREPS)*WCROS(2,ISLICE,ISUP) NTARAYT 4821
Z *(1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)*CPC(ISUP)/ NTARAYT 4822
Z WXCP - HX*(1.-TREPS)*HC(ISUP)*(YCONVU+ NTARAYT 4823
Z YPINSU)*(A1+A3)/WXCP - TRTRMG) NTARAYT 4824
Z - T(1,ISLICE,LIN)*HX*(1.-TREPS)*HC(IS)*A2/WXCP NTARAYT 4825
Z - T(1,ISLICE,LCOOL)*((1.-TREPS)*((-FILMW*144.*3600.*CPC(IS)/ NTARAYT 4826
Z WXCP)-1.0-(GAMC(IS)-1.)/2.)*AM2(IS) NTARAYT 4827
Z - HX*HC(IS)*(A2+A4)*(YCONV+YPINS)/WXCP - TRTRMG) - TREDGE NTARAYT 4828
IF (IS.GT.NFWD) TCOF(LCOOL,J6) = TREPS*HX*HC(ISUP)*A3/WXCP NTARAYT 4829
IF (IS.GT.NFWD) TCOF(LCOOL,J8) = TREPS*HX*HC(IS)*A4/WXCP NTARAYT 4830
C
C*** OF THE TERMS YIMP,YPINS,YCONV, ONLY ONE CAN BE NON-ZERO AT A TIME
C YIMP = 1.0 MEANS THAT IMPINGEMENT HEAT TRANSFER IS BEING CONSIDEREDNTARAYT 4831
C YPINS = 1.0 MEANS THAT A PIN FINNED SURFACE IS BEING USED NTARAYT 4832
C YCONV = 1.0 MEANS A FORCED CONVECTION CORRELATION IS BEING USED NTARAYT 4833
C
IF (TYME.GT.0.) RCHRD = (144./3600.)*AKW(ISLICE,IS)*DLTYME/
Z (RHOM*SPHTM*((DX1+DX2)/2.)**2)) NTARAYT 4834
IF (TYME.GT.0.) RTRNV = (144./3600.)*AKW(ISLICE,IS)*DLTYME/
Z (RHOM*SPHTM*(TAU(L)**2)) NTARAYT 4835
IF (RCHRD.GT.RCHRD) RCHRD = RCHRD NTARAYT 4836
IF (RTRNV.GT.RTRNVM) RTRNVM = RTRNV NTARAYT 4837
GO TO 430 NTARAYT 4838

```

350 CONTINUE
 C FOR MIXING ZONE; STATION NO. NFWD+1:
 C
 IF (IHC(IS).EQ.3) CALL HCPINS (IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINSNTARYT 4861
 Z AHTRN1 = EFAREA (IS),EFAREA)
 IF (IHC(IS).EQ.3) A2 = EFAREA (IS)/2.
 A3 = SPAN*DLX(LCOOLP)/2.
 IF (IHC(IS-1).EQ.3) A3 = EFAREA (IS-1)/2.
 A4 = SPAN*DLX(LCOOLP)/2.
 IF (IHC(IS).EQ.3) A4 = EFAREA (IS+1)/2.
 IF (IHC(IS).EQ.3) GO TO 360
 C
 IF (IHC(IS).EQ.2) HC (IS) = HCFRCD (IS,LCOOL,LIN)
 AHTRN1 = (DX5+DX6)*S (ISLICE)/2.
 360 CONTINUE
 C
 IF (IHC(IS-1).EQ.1) YIMPUU = 1.0
 IF (IHC(IS-1).EQ.2) YCNVUU=1.0+RCVRY*AM2 (IS-1)*(GAMC (IS-1)-1.)/2.
 IF (IHC(IS-1).EQ.3) YFNSUU = 1.0
 C
 WXCP = WCROS(2,ISLICE,IS)*144.*CPC (IS)*3600.
 RHOBAR = ((P(1,ISLICE,IS-2)+P(1,ISLICE,IS))/24.)*(1./R)*2.*
 Z WCROS(2,ISLICE,IS)/
 Z (T(1,ISLICE,LCUPP+1)*WCROS(2,ISLICE,IS-1) + T(1,ISLICE,LCUP)* NTARAYT 4875
 Z WCROS(2,ISLICE,IS-2)
 Z + TOG*WDUMP + T(1,ISLICE,LCOOL)*WCROS(2,ISLICE,IS)) NTARAYT 4876
 VOLBAR = (A(LCUP)+A(LCUPP+1)+A(LCOOL))*(DLX(LCOOLP)+DLX(LIN))/4. NTARAYT 4877
 TRTRMJ = 0.0
 IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMJ = PHOBAR*VOLBAR/
 Z (2.*DLTYME*(WCROS(2,ISLICE,IS)**2)) NTARAYT 4878
 TCOF(LCOOL,1) = TREPS*HX*HC (IS-2)*A1/WXCP NTARAYT 4879
 TCOF(LCOOL,2) = TREPS*((WCROS(2,ISLICE,ISUP)/WCROS(2,ISLICE,IS)) NTARAYT 4880
 Z *(1.+AM2 (ISUP)*(GAMC (ISUP)-1.)/2.)*(CPC (ISUP)/ NTARAYT 4881
 Z CPC (IS)) - HX*HC (ISUP)*(YCONVU+YPINSU)*(A1)/WXCP NTARAYT 4882
 Z - TRTRMJ*WCROS(2,ISLICE,IS-2) NTARAYT 4883
 TCOF(LCOOL,6) = TREPS*HX*HC (IS-1)*A3/WXCP NTARAYT 4884
 TCOF(LCOOL,7) = TREPS*((WCROS(2,ISLICE,IS-1)/WCROS(2,ISLICE,IS))* NTARAYT 4885
 Z (1.+AM2 (IS-1)*(GAMC (IS-1)-1.)/2.)*(CPC (IS-1)/ NTARAYT 4886
 Z CPC (IS)) - HX*HC (IS-1)*(YCNVUU+YFNSUU)*(A3)/WXCP NTARAYT 4887
 Z - TRTRMJ*WCROS(2,ISLICE,IS-1) NTARAYT 4888
 TCOF(LCOOL,11) = TREPS*HX*HC (IS)*A2/WXCP NTARAYT 4889
 TCOF(LCOOL,12)=TREPS*((-PILMW/WCROS(2,ISLICE,IS))-1.0-((GAMC (IS)
 Z -1.)/2.)*AM2 (IS)-HX*HC (IS)*(A2+A4)*(YCONV+YPINS)
 Z /WXCP) - TRTRMJ*WCROS(2,ISLICE,IS) NTARAYT 4890
 TCOF(LCOOL,16) = TREPS*HX*HC (IS)*A4/WXCP NTARAYT 4891
 TCOF(LCOOL,24) = TOG*(HX*HC (IS-2)*A1*YIMPU + HX*HC (IS)*(A2+A4)*
 Z YIMP + HX*HC (IS-1)*A3*YIMPUU)/WXCP NTARAYT 4892
 Z - PUMP (IS)/(CPC (IS)*778.*144.*32.2) - (CPO/CPC (IS))*TOG* NTARAYT 4893
 Z (WJ (ISLICE,IS-2)+WJ (ISLICE,IS-1)+WDUMP)/WCROS(2,ISLICE,IS) NTARAYT 4894
 Z - T(1,ISLICE,LCUPP)*(1.-TREPS)*HX*HC (IS-2)*A1/WXCP NTARAYT 4895
 Z - T(1,ISLICE,LCUP)*(1.-TREPS)*((WCROS(2,ISLICE,ISUP)/
 Z WCROS(2,ISLICE,IS))*(1.+AM2 (ISUP)*(GAMC (ISUP)-1.)/2.) NTARAYT 4896
 Z *(CPC (ISUP)/CPC (IS)) - HX*HC (ISUP)*(YCONVU+YPINSU)
 Z *(A1)/WXCP) + TRTRMJ*WCROS(2,ISLICE,IS-2)) NTARAYT 4897
 Z - T(1,ISLICE,LCUPP)*(1.-TREPS)*HX*HC (IS-1)*A3/WXCP NTARAYT 4898
 Z - T(1,ISLICE,LCUPP+1)*((1.-TREPS)*((WCROS(2,ISLICE,IS-1)/
 Z WCROS(2,ISLICE,IS))*(1.+AM2 (IS-1)*(GAMC (IS-1)-1.)/2.) NTARAYT 4899
 Z *(CPC (IS-1)/CPC (IS)) - HX*HC (IS-1)*(YCNVUU+YFNSUU)*(A3)/ NTARAYT 4900


```

C NTARAYT 4981
C NTARAYT 4982
C NTARAYT 4983
C NTARAYT 4984
C NTARAYT 4985
C NTARAYT 4986
C NTARAYT 4987
C*****NTARAYT 4988
C*****NTARAYT 4989
C***** SETUP FOR TRAILING EDGE REGION: NTARAYT 4990
C*****NTARAYT 4991
C*****NTARAYT 4992
380 CONTINUE NTARAYT 4993
ISENS = IS - 2*(IS/2) NTARAYT 4994
C NTARAYT 4995
C ISENS = 0 MEANS IS IS EVEN AND STATION IS ON SUCTION SIDE NTARAYT 4996
C ISENS = 1 MEANS IS IS ODD AND STATION IS ON PRESSURE SIDE NTARAYT 4997
C NTARAYT 4998
LCOOL = 5*IS NTARAYT 4999
LIN = LCOOL - 1 NTARAYT 5000
L = LCOOL - 2 NTARAYT 5001
LJ = LCOOL - 3 NTARAYT 5002
LOUT = LCOOL - 4 NTARAYT 5003
LUP = L - 10 NTARAYT 5004
LDN = L + 10 NTARAYT 5005
LCUP = LCOOL - 10 NTARAYT 5006
LCUPS = LCUP - 1 NTARAYT 5007
LCUPP = LCOOL - 6 NTARAYT 5008
LCOOLP = LCOOL + 4 NTARAYT 5009
C NTARAYT 5010
C NTARAYT 5011
I3 = 12 - LIN + LCOOL NTARAYT 5012
J1 = 12 - L + LUP NTARAYT 5013
J2 = 12 - L + LDN NTARAYT 5014
J4 = 12 - LCOOL + LCUPS NTARAYT 5015
J5 = 12 - LCOOL + LCUP NTARAYT 5016
J6 = 12 - LCOOL + LCUPP NTARAYT 5017
J8 = 12 - LCOOL + LCOOLP NTARAYT 5018
J9 = 16 NTARAYT 5019
C NTARAYT 5020
A1 = SPAN*DLX(LIN)/2. NTARAYT 5021
A2 = A1 NTARAYT 5022
A3 = SPAN*DLX(LCOOLP)/2. NTARAYT 5023
A4 = A3 NTARAYT 5024
IF (IHC(IS-2).EQ.3) A1 = ZPAREA(IS-2)/2. NTARAYT 5025
IF (IHC(IS-2).EQ.3) A3 = EFAREA(IS-1)/2. NTARAYT 5026
DX1 = DLX(L) NTARAYT 5027
DX2 = DLX(LDN) NTARAYT 5028
DX3 = DLX(LOUT) NTARAYT 5029
DX4 = DLX(LDN-2) NTARAYT 5030
DX5 = DLX(LIN) NTARAYT 5031
DX6 = DLX(LDN+1) NTARAYT 5032
DX7 = DLX(LCOOL) NTARAYT 5033
DX9 = DLX(LJ) NTARAYT 5034
DX10= DLX(LDN-1) NTARAYT 5035
C NTARAYT 5036
IF (IHC(ISUP).EQ.1) YIMPU = 1.0 NTARAYT 5037
IF (IHC(ISUP).EQ.2) YCONVU=1.0+RCVRY*AM2(ISUP)*(GAMC(ISUP)-1.)/2. NTARAYT 5038
IF (IHC(ISUP).EQ.3) YFINSU = 1.0 NTARAYT 5039
C NTARAYT 5040

```

```

IF (IS.LT.NSTA-1) GO TO 390          NTA RAYT 5041
C                                     NTA RAYT 5042
C FOR THE LAST STATIONS IN THE TRAILING EDGE:  NTA RAYT 5043
C                                     NTA RAYT 5044
DX2 = 0.0                           NTA RAYT 5045
DX4 = 0.0                           NTA RAYT 5046
DX6 = 0.0                           NTA RAYT 5047
DX10= 0.0                          NTA RAYT 5048
390 CONTINUE                         NTA RAYT 5049
IF ( IHC(IS).EQ.3) GO TO 420         NTA RAYT 5050
IF ( IHC(IS).EQ.2) GO TO 410         NTA RAYT 5051
GO TO 340                           NTA RAYT 5052
C                                     NTA RAYT 5053
C*** HCPFCD COMPUTES HC FOR FORCED CONVECTION  NTA RAYT 5054
C                                     NTA RAYT 5055
410 CONTINUE                         NTA RAYT 5056
TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.  NTA RAYT 5057
CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)        NTA RAYT 5058
RE(IS) = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)  NTA RAYT 5059
C                                     NTA RAYT 5060
HC(IS) = .023*12.* (C/DH(IS))* (RE(IS)**.8)*(PD**.333)  NTA RAYT 5061
AHTRN1 = (DX5 + DX6)*SPAN/2.           NTA RAYT 5062
GO TO 340                           NTA RAYT 5063
C                                     NTA RAYT 5064
C***** SUBROUTINE HCPINS COMPUTES HC FOR A PIN FIN SURFACE OR FOR  NTA RAYT 5065
C TURBULENT FORCED CONVECTION CHANNEL FLOW  NTA RAYT 5066
420 CONTINUE                         NTA RAYT 5067
IF(ISENS.EQ.0) GO TO 424             NTA RAYT 5068
AHTRN1 = EFAREA(IS)                 NTA RAYT 5069
HC(IS) = HC(IS-1)                   NTA RAYT 5070
GO TO 340                           NTA RAYT 5071
C                                     NTA RAYT 5072
424 CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)  NTA RAYT 5073
AHTRN1 = EFAREA(IS)                 NTA RAYT 5074
IF (IS.GE.NSTA-1) AHTRN1 = AHTRN1/2.  NTA RAYT 5075
A2 = EFAREA(IS)/2.                  NTA RAYT 5076
A4 = EFAREA(IS+1)/2.                NTA RAYT 5077
GO TO 340                           NTA RAYT 5078
C                                     NTA RAYT 5079
C                                     NTA RAYT 5080
C                                     NTA RAYT 5081
C                                     NTA RAYT 5082
430 CONTINUE                         NTA RAYT 5083
440 CONTINUE                         NTA RAYT 5084
450 CONTINUE                         NTA RAYT 5085
C                                     NTA RAYT 5086
RETURN                               NTA RAYT 5087
END                                  NTA RAYT 5088

```

```

C---- SOURCE.NTCOFTT
      SUBROUTINE TCOEP(IWRITE,WS,NIT,IPLOT,ALPH2)          NTCOFTT 5089
C                                     NTCOFTT 5090
C- SOURCE.NTCOFTT                      NTCOFTT 5091
C                                     NTCOFTT 5092
C                                     NTCOFTT 5093
      DIMENSION POLD(15,80), PSAV(5), X(80), ALPH2(4), DELTAN(15),  NTCOFTT 5094
      TTOTC(80), JSO(15)                                NTCOFTT 5095
      DIMENSION PEXOLD(15), PIMOLD(15)                  NTCOFTT 5096
      REAL*8 TCOP                                         NTCOFTT 5097
      COMMON /MATRIX/ TCOP(400,30)                      NTCOFTT 5098
C                                     NTCOFTT 5099
      COMMON /PRPS/ CPO, GAM0, DP(80), SP(80), RE(80),  NTCOFTT 5100

```

Z	CPC(80),	GAMC(80),	DUMR1(80),	DUMR2(80)	NTCOFTT 5101	
C					NTCOFTT 5102	
C					NTCOFTT 5103	
Z	COMMON /TCO/ ADUMP,	BTA,	CD,	CP,	NTCOFTT 5104	
Z	GAM,	PIM,	R,	SPAN,	TOG,	NTCOFTT 5105
Z	WDUMP,	WIM,	AKC(15,80),	AKW(15,80),	NTCOFTT 5106	
Z	A(400),	AJET(80),	AM2(80),	CNUM(80),	NTCOFTT 5107	
Z	DH(80),	DHF(80),	DHJ(80),		NTCOFTT 5108	
Z	DLX(400),	FF(80),	HC(80),	HG(80),	NTCOFTT 5109	
Z	P(2,15,80),PEXIT(15),		PUMP(80),	QG(80),	NTCOFTT 5110	
Z	QSNK(80),	RR(80),	S(15),	T(2,15,400),	NTCOFTT 5111	
Z	TG(80),	TAU(400),	WPC(80),	XN(80),	NTCOFTT 5112	
Z	WJ(15,80),	WCROS(2,15,80),		ITIP,	NTCOFTT 5113	
Z	ICOR,	IFILM,	IHUB,		NTCOFTT 5114	
Z	ISBLOK,	ISLICE,	NBLKSZ,	NSLICE,	NTCOFTT 5115	
Z	NFWD,	NSTA,	IHC(80)		NTCOFTT 5116	
C					NTCOFTT 5117	
Z	COMMON /TRNSNT/ RHOC,	RHOM,	SPHTC,	SPHTM,	NTCOFTT 5118	
Z	DLTYME,	TYME,	TEPS,	TYMMAX	NTCOFTT 5119	
C					NTCOFTT 5120	
C					NTCOFTT 5121	
C	PROGRAM IS SET TO WORK WITH THE FOLLOWING UNITS ON THE VARIABLES:				NTCOFTT 5122	
C	PRESSURE, P, IS IN PSIA *****				NTCOFTT 5123	
C	FLOW RATES, WC,WJ,&WCROS ARE IN LBM/SEC *****				NTCOFTT 5124	
C	AREAS ARE IN IN**2 *****				NTCOFTT 5125	
C	ALL LENGTHS ARE IN INCHES *****				NTCOFTT 5126	
C	HEAT TRANSFER COEFFICIENTS IN BTU/(HR*FT**2*R) *****				NTCOFTT 5127	
C	BTA = 0. INDICATES THAT A HEAT TRANSFER COEFFICIENT BOUNDARY				NTCOFTT 5128	
C	CONDITION IS SPECIFIED				NTCOFTT 5129	
C	1. INDICATES THAT A HEAT FLUX IS SPECIFIED ON THE GAS BOUNDARY				NTCOFTT 5131	
C	UNITS EXPECTED ON THE FOLLOWING INPUT DATA ARE:				NTCOFTT 5132	
C	LENGTHS ARE ALL IN INCHES-- DLX, TAU, SPAN, DH, DPX				NTCOFTT 5133	
C	TEMPERATURES ARE ABSOLUTE (R)				NTCOFTT 5134	
C	MASS FLOWS ARE ALL IN (LBM/SEC),--- WCROS, WJ, WPC, WDUMP				NTCOFTT 5135	
C	HEAT TRANSFER COEFFICIENTS ARE IN BTU/(HR*FT**2*R),--HG,HC--GAS				NTCOFTT 5136	
C	SIDE AND COOLANT SIDE .				NTCOFTT 5137	
C	HEAT FLUX QG IS IN BTU/(HR*FT**2)				NTCOFTT 5138	
C	HEAT SINK, QSNK, IS IN BTU/(HR)				NTCOFTT 5139	
C	THERMAL PROPERTIES ARE: CONDUCTIVITY-AKW-BTU/(HR*FT*R)				NTCOFTT 5140	
C	HEAT CAPACITY-CP-BTU/(LBM*R)				NTCOFTT 5141	
C					NTCOFTT 5142	
C					NTCOFTT 5143	
C	----SET UP FIRST GUESS AT TEMPERATURE DISTRIBUTION				NTCOFTT 5144	
C	----ASSUME COOLANT TEMPERATURE IS CONSTANT, = PLENUM STATIC TEMPERATURE				NTCOFTT 5145	
C	----ASSUME METAL TEMPERATURE IS CONSTANT, = 2200. R				NTCOFTT 5146	
C					NTCOFTT 5147	
100	CONTINUE				NTCOFTT 5148	
	NODST = 5*NSTA				NTCOFTT 5149	
	N = NSTA - 1				NTCOFTT 5150	
	V = .70				NTCOFTT 5151	
	NSAVE = NODST-10				NTCOFTT 5152	
	TSAVE = T(2,ISLICE,NSAVE)				NTCOFTT 5153	
C					NTCOFTT 5154	
C	NODST = NODE NUMBER OF LAST FLOW CHANNEL NODE,AT EXIT OF TRAILING EDG				NTCOFTT 5155	
C					NTCOFTT 5156	
110	TMP = TOG				NTCOFTT 5157	
	NODSP = 5*NFWD				NTCOFTT 5158	
120	CONTINUE				NTCOFTT 5159	
C					NTCOFTT 5160	

```

C-- FOR TRANSIENT CASES, ADJUST INITIAL GUESS OF PRESSURE DISTRIBUTION NTCOPTT 5161
C-- BASED ON THE VARIATION OF SUPPLY AND EXIT PRESSURES. NTCOPTT 5162
C NTCOPTT 5163
C IF (TYME.GT.0.0) GO TO 234 NTCOPTT 5164
C NTCOPTT 5165
C-- FOR STEADY STATE, ONLY INITIALIZE P'S ON FIRST OVERALL LOOP. NTCOPTT 5166
C NTCOPTT 5167
C IF (NIT.GT.1) GO TO 255 NTCOPTT 5168
PEXOLD(ISLICE) = PEXIT(ISLICE) NTCOPTT 5169
PIMOLD(ISLICE) = PIM NTCOPTT 5170
160 WRITE(6,165) (I,T(1,ISLICE,I),I=1,NODST) NTCOPTT 5171
165 FORMAT(1H1,' ASSUMED INITIAL TEMPERATURE DISTRIBUTION, (NODE NO.',NTCOFTT 5172
Z ,',T) '/7(' (',I3,',',P8.2,')')) NTCOPTT 5173
170 CONTINUE NTCOPTT 5174
C NTCOPTT 5175
C----PRESSURE INITIALIZATION, GIVEN PIM (PLENUM STATIC PRESSURE) AND NTCOPTT 5176
C----PEXIT (GAS SIDE STATIC PRESSURE AT TRAILING EDGE), FIT PRESSURE TO NTCOPTT 5177
C A NTCOPTT 5178
C----CUBIC EQN. OF THE FORM A+B*X**3, ASSUMING 85% OF THE PRESSURE DROP NTCOPTT 5179
C----OCCURS IN THE TRAILING EDGE CHANNEL NTCOPTT 5180
C NTCOPTT 5181
X(1) = 0.0 NTCOPTT 5182
X(2) = DLX(10) NTCOPTT 5183
DO 180 I = 3,NFWD NTCOPTT 5184
X(I) = DLX(5*I) + X(I-2) NTCOPTT 5185
180 CONTINUE NTCOPTT 5186
P(1,ISLICE,1) = PIM - (PIM-PEXIT(ISLICE))* .15 NTCOPTT 5187
P(1,ISLICE,N) = PEXIT(ISLICE) NTCOPTT 5188
P(1,ISLICE,N+1) = P(1,ISLICE,N) NTCOPTT 5189
P(1,ISLICE,NFWD-1)=P(1,ISLICE,1)-.2*(P(1,ISLICE,1)-PEXIT(ISLICE)) NTCOPTT 5190
190 P(1,ISLICE,NFWD) = P(1,ISLICE,NFWD-1) NTCOPTT 5191
DO 200 I = 3,NFWD,2 NTCOPTT 5192
P(1,ISLICE,I) = P(1,ISLICE,1) - (P(1,ISLICE,1)-P(1,ISLICE,NFWD))* NTCOPTT 5193
Z (X(I)/X(NFWD))**3 NTCOPTT 5194
200 P(1,ISLICE,I-1)=P(1,ISLICE,1)-(P(1,ISLICE,1)-P(1,ISLICE,NFWD-1))* NTCOPTT 5195
Z (X(I-1)/X(NFWD-1))**3 NTCOPTT 5196
ISTR = NFWD+1 NTCOPTT 5197
IFNL = N NTCOPTT 5198
C NTCOPTT 5199
C----POE TRAILING EDGE CHANNEL, X VALUES ARE RELATIVE TO END OF NTCOPTT 5200
C IMPINGEMENT CHANNEL NTCOPTT 5201
C NTCOPTT 5202
X(NFWD+1) = (DLX(NODSF) + DLX(NODSF-5))/2. NTCOPTT 5203
210 ITEM = NFWD+3 NTCOPTT 5204
DO 220 I = ITEM,N,2 NTCOPTT 5205
LCOOL = 5*I NTCOPTT 5206
X(I) = X(I-2) + DLX(LCOOL) NTCOPTT 5207
220 CONTINUE NTCOPTT 5208
DO 230 I = ISTR,IFNL,2 NTCOPTT 5209
P(1,ISLICE,I) = P(1,ISLICE,NFWD)-(P(1,ISLICE,NFWD)-PEXIT(ISLICE))* NTCOPTT 5210
Z (X(I)/X(N))**3 NTCOPTT 5211
230 P(1,ISLICE,I+1) = P(1,ISLICE,I) NTCOPTT 5212
DO 232 I = 1,NSTA NTCOPTT 5213
232 POLD(ISLICE,I) = P(1,ISLICE,I) NTCOPTT 5214
DO 233 I = 1,NSTA NTCOPTT 5215
233 P(2,ISLICE,I) = P(1,ISLICE,I) NTCOPTT 5216
GO TO 240 NTCOPTT 5217
C NTCOPTT 5218
234 DLTAPC = .84*(PIM-PIMOLD(ISLICE)) NTCOPTT 5219
PIMOLD(ISLICE) = PIM NTCOPTT 5220

```

```

DLTAPE = PEXIT(ISLICE)-PEXOLD(ISLICE)          NTCOFTT 5221
PEXOLD(ISLICE) = PEXIT(ISLICE)                  NTCOFTT 5222
DO 235 I = 1,NFWD          NTCOFTT 5223
235   P(2,ISLICE,I) = P(2,ISLICE,I) + DLTAPE    NTCOFTT 5224
      ISTRT = NFWD+1          NTCOFTT 5225
      IFNL = NSTA-1          NTCOFTT 5226
      DO 236 I = ISTRT,IFNL,2 NTCOFTT 5227
      P(2,ISLICE,I) = P(2,ISLICE,I) + DLTAPE*(1.0-X(I)/X(IFNL)) + NTCOFTT 5228
                                         DLTAPE*X(I)/X(IFNL) NTCOFTT 5229
Z
236   P(2,ISLICE,I+1) = P(2,ISLICE,I)          NTCOFTT 5230
      DO 237 I = 1,NSTA          NTCOFTT 5231
237   POLD(ISLICE,I) = P(2,ISLICE,I)          NTCOFTT 5232
      GO TO 255          NTCOFTT 5233
C
240   WRITE(6,245)          NTCOFTT 5234
245   FORMAT(/' INITIAL PRESSURE DIST. (STATION NO.,P)  '/)
      WRITE(6,250) (I,P(1,ISLICE,I),I=1,ISTRT) NTCOFTT 5235
      WRITE(6,250) (I,P(1,ISLICE,I),I=ITEM,N,2) NTCOFTT 5236
250   FORMAT(7(' (',I3,',',F7.2,')'))          NTCOFTT 5237
C
C
C
255   CONTINUE          NTCOFTT 5238
      DO 260 I=1,4          NTCOFTT 5239
260   PSAV(I)=0.0          NTCOFTT 5240
C
C
C
C----INITIALLY, THE FLOW SPLIT AT THE LEADING EDGE IS ASSUMED
C      TO BE 50/50 (DELTA=.5)          NTCOFTT 5241
C
C----IDEKT COUNTS THE NUMBER OF FLOW SPLIT ITERATIONS. IF NO
C      CONVERGENCE, IDELT IS SET NEGATIVE.          NTCOFTT 5242
C----DELTAN IS THE FRACTION OF FLCW TO THE SUCTION SIDE (EVEN
C      NUMBERED STATIONS)          NTCOFTT 5243
C----IVERGE COUNTS THE NUMBER OF ITERATIONS AT A GIVEN FLOW SPLIT
C
C IFNL = THE NUMBER OF FLOW CHANNEL NODES, USED IN PRESSURE CALCULATIONS NTCOFTT 5244
C
275   IF (NIT.EQ.1.AND.TYME.LT.0.0) DELTAN(ISLICE) = .5          NTCOFTT 5245
      CONTINUE          NTCOFTT 5246
      IFNL = NSTA - 3          NTCOFTT 5247
      IVERGE = 0          NTCOFTT 5248
      IDELT = 1          NTCOFTT 5249
      JS = 1          NTCOFTT 5250
      IF (NIT.GT.1) JS = JS0(ISLICE)          NTCOFTT 5251
290   CONTINUE          NTCOFTT 5252
      IVERGE = IVERGE + 1          NTCOFTT 5253
300   CONTINUE          NTCOFTT 5254
C
      JSENS = JS - 2*(JS/2)          NTCOFTT 5255
C
C----SUBROUTINE FLOWS COMPUTES JET FLOW RATES, CROSSFLOW RATES, AND
C      THE SQUARE OF THE MACH NUMBER          NTCOFTT 5256
C
310   CONTINUE          NTCOFTT 5257
C
      CALL FLOWS(JS,DELTAN,ICHOKE,AMCHOK)          NTCOFTT 5258
320   CONTINUE          NTCOFTT 5259
      IF (WJ(ISLICE,JS).LE.0.0) GO TO 370          NTCOFTT 5260
C
C SUBROUTINE HCOOL COMPUTES IMPINGEMENT REGION HEAT TRANSFER COEFF'NTS NTCOFTT 5261

```

```

C
      CALL HCOOL(JS)
330  CONTINUE
      CALL THRCOM
C SUBROUTINE THRCOM EXTRACTS THERMAL CONDUCTIVITIES FROM INPUT
C TABLES AKCTBL AND AKWTBL.
C
335  CONTINUE
C
C----> CHECK TO MAKE SURE STAGNATION HC IS LESS THAN MAXIMUM PHYSICALLY
C      POSSIBLE VALUE.
C
IF (JS.EQ.1) ASTG = (DLX(9)+DLX(14))*SPAN/2.0
IF (JS.GT.1) ASTG = (DLX(5*JS-1)+DLX(5*JS+9))*SPAN/2.0
HSTGMX = WJ(ISLICE,JS)*CFO*144.*3600./ASTG
IF (HC(JS).GT.HSTGMX) HC(JS) = HSTGMX
C
C
337  CONTINUE
      CALL TAPPAY(JS,JSENS,DELTAN)
C
C SOLVE THE TCOF ARRAY AND SET NEW TEMPERATURE VALUES:
C
340  CONTINUE
      CALL GAUSS(NODST,23)
      DO 350 I = 1,NODST
      T(2,ISLICE,I) = TCOF(I,24)
      IF (T(2,ISLICE,I).LE.0.0) T(2,ISLICE,I) = T0G
350  CONTINUE
C***** *****
360  IF (ABS((T(2,ISLICE,NSAVE)-TSAVE)/TSAVE).GT..05)
      Z          CALL FLOWS(JS,DELTAN,ICHOK,AMCHOK)
      TSAVE = T(2,ISLICE,NSAVE)
C***** *****
      IF (ICHOK.EQ.0) GO TO 370
      WRITE(8,365) ISLICE,IVERGE,IDELET,NIT,ICHOK,AMCHOK
365  FORMAT(/10X,'SLICE ',I2,' IS CHOKE FOR IVERGE =',I3,', IDELT =',
      Z     I3,', NIT =',I3,', ICHOK =',I4,', M**2 =',F6.3)
370  CONTINUE
C
C COMPUTE NEW PRESSURES
C
      CALL PARRAY(JS,JSENS,ICHOK)
C
C SOLVE THE TCOF ARRAY AND COMPUTE NEW PRESSURES
C
430  CONTINUE
      CALL GAUSS(IFNL,19)
440  CONTINUE
      DO 460 I = 1,IPNL
450  P(2,ISLICE,I) = TCOF(I,20)
460  CONTINUE
      P(2,ISLICE,IFNL+1) = P(2,ISLICE,IFNL)
      P(2,ISLICE,NSTA-1) = PEXIT(ISLICE)
      P(2,ISLICE,NSTA) = PEXIT(ISLICE)
470  CONTINUE
      IF (IWRITE.EQ.2) CALL WROUT(IDELET,JS,DELTAN,IVERGE)
      IF (IPLOT.EQ.2) CALL PLOTMF(ALPH2)
C
480  CONTINUE

```

NTCOFTT 5281
 NTCOFTT 5282
 NTCOFTT 5283
 NTCOFTT 5284
 NTCOFTT 5285
 NTCOFTT 5286
 NTCOFTT 5287
 NTCOFTT 5288
 NTCOFTT 5289
 NTCOFTT 5290
 NTCOFTT 5291
 NTCOFTT 5292
 NTCOFTT 5293
 NTCOFTT 5294
 NTCOFTT 5295
 NTCOFTT 5296
 NTCOFTT 5297
 NTCOFTT 5298
 NTCOFTT 5299
 NTCOFTT 5300
 NTCOFTT 5301
 NTCOFTT 5302
 NTCOFTT 5303
 NTCOFTT 5304
 NTCOFTT 5305
 NTCOFTT 5306
 NTCOFTT 5307
 NTCOFTT 5308
 NTCOFTT 5309
 NTCOFTT 5310
 NTCOFTT 5311
 NTCOFTT 5312
 NTCOFTT 5313
 NTCOFTT 5314
 NTCOFTT 5315
 NTCOFTT 5316
 NTCOFTT 5317
 NTCOFTT 5318
 NTCOFTT 5319
 NTCOFTT 5320
 NTCOFTT 5321
 NTCOFTT 5322
 NTCOFTT 5323
 NTCOFTT 5324
 NTCOFTT 5325
 NTCOFTT 5326
 NTCOFTT 5327
 NTCOFTT 5328
 NTCOFTT 5329
 NTCOFTT 5330
 NTCOFTT 5331
 NTCOFTT 5332
 NTCOFTT 5333
 NTCOFTT 5334
 NTCOFTT 5335
 NTCOFTT 5336
 NTCOFTT 5337
 NTCOFTT 5338
 NTCOFTT 5339
 NTCOFTT 5340

```

C NTCOFTT 5341
C CHECK OVERALL CONVERGENCE NTCOFTT 5342
C CALCULATIONS ARE REPEATED UNTIL THE PRESSURE AT STATION 1 (NODE 5) NTCOFTT 5343
C HAS STABILIZED FOR NTCOFTT 5344
C FOUR ITERATIONS. THEN WE GO TO CHECK THE FLOW SPLIT. NTCOFTT 5345
C NTCOFTT 5346
C NTCOFTT 5347
DO 490 I=1,3 NTCOFTT 5348
K=5-I NTCOFTT 5349
490 PSAV(K)=PSAV(K-1) NTCOFTT 5350
PSAV(1)=P(2,ISLICE,JS) NTCOFTT 5351
C NTCOFTT 5352
DIFO=0.0 NTCOFTT 5353
DO 500 I=1,3 NTCOFTT 5354
JJ=I+1 NTCOFTT 5355
DO 500 K=JJ,4 NTCOFTT 5356
DIFN=ABS(PSAV(I)-PSAV(K)) NTCOFTT 5357
IF(DIFO.LT.DIFN)DIFO=DIFN NTCOFTT 5358
500 CONTINUE NTCOFTT 5359
C NTCOFTT 5360
510 DIFO=DIFO/(PIM-PEXIT(ISLICE)) NTCOFTT 5361
PCNVRG = .01 NTCOFTT 5362
IF (NIT.EQ.1.AND.NSLICE.GT.1) PCNVRG = .05 NTCOFTT 5363
EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NPWD))/(P(2,ISLICE,NPWD-1)) NTCOFTT 5364
IF (IDELT.EQ.1.AND.IVERGE.LT.8.AND.TYME.LE.0.0) GO TO 516 NTCOFTT 5365
IF (DIFO.LE.PCNVRG.AND.IVERGE.GE.4) GO TO 560 NTCOFTT 5366
516 CONTINUE NTCOFTT 5367
IF (IVERGE.LT.10) GO TO 520 NTCOFTT 5368
IF (ABS(PSAV(1)-PSAV(2)).GT.ABS(PSAV(3)-PSAV(4))) V=1.-(1.-V)/2. NTCOFTT 5369
520 CONTINUE NTCOFTT 5370
DO 530 I = 1,NSTA NTCOFTT 5371
P(2,ISLICE,I) = P(2,ISLICE,I) + V*(POLD(ISLICE,I) - P(2,ISLICE,I)) NTCOFTT 5372
IF (P(2,ISLICE,I).GE.PIM) P(2,ISLICE,I) = .999*PIM NTCOFTT 5373
POLD(ISLICE,I) = P(2,ISLICE,I) NTCOFTT 5374
TTOTC(I) = T(2,ISLICE,5*I)*(1.+(GAMC(I)-1.)*AM2(I)/2.) NTCOFTT 5375
530 CONTINUE NTCOFTT 5376
C NTCOFTT 5377
540 CONTINUE NTCOFTT 5378
IF (IVERGE.GT.30.OR.V.GT..95) NTCOFTT 5379
Z WRITE(8,550) IVERGE,V,(PSAV(I),I=1,4) NTCOFTT 5380
550 FORMAT(' ***** CONVERGENCE PROBLEMS *****', NTCOFTT 5381
Z ' IVERGE=',I3,'; V= ',F6.4,'; PSAV= ',4(F10.2)) NTCOFTT 5382
IP (IVERGE.GT.50) GO TO 590 NTCOFTT 5383
GO TO 290 NTCOFTT 5384
C NTCOFTT 5385
***** ONCE PRESSURE-TEMPERATURE ITERATION HAS CONVERGED, CHECK
C THE FLOW SPLIT AND ADJUST IF NECESSARY. NTCOFTT 5386
C NTCOFTT 5387
560 CONTINUE NTCOFTT 5388
IF ( IWRITE.EQ.1 ) CALL WROUT(IDELT,JS,DELTAN,IVERGE) NTCOFTT 5389
IP (ICHOKE.GT.0) WRITE(6,565) ICHOKE,AMCHOK NTCOFTT 5390
565 FORMAT(/10X,'MACH NO. > 1 AT STATION ',I4,', M**2 = ',F6.3/) NTCOFTT 5391
570 CONTINUE NTCOFTT 5392
EPSN = (P(2,ISLICE,NPWD-1)-P(2,ISLICE,NPWD))/(P(2,ISLICE,NPWD-1)) NTCOFTT 5393
575 CONTINUE NTCOFTT 5394
IF (TYME.GT.0.0) GO TO 590 NTCOFTT 5395
CALL FLSPLT(AJET,EPSN,ISLICE,NODSP,IDEKT,JS,DELTAN,ICONV) NTCOFTT 5396
580 CONTINUE NTCOFTT 5397
IF (ICONV.EQ.1) CALL WROUT(IDELT,JS,DELTAN,IVERGE) NTCOFTT 5398
IVERGE = 0 NTCOFTT 5399
IF (ICONV.EQ.0) GO TO 290 NTCOFTT 5400

```

```

590 CONTINUE NTCOPTT 5401
IP (TIME.GT.0.0) CALL WROUT(IDELT,JS,DELTAN,IVERGE)
JSO(ISLICE) = JS NTCOPTT 5402
RETURN NTCOPTT 5403
END NTCOPTT 5404
NTCOPTT 5405

C----SOURCE.NTHRCNT NTHRCNT 5406
SUBROUTINE THRCON NTHRCNT 5407
C NTHRCNT 5408
C SOURCE.NTHRCNT---- NTHRCNT 5409
C NTHRCNT 5410
COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGF(1000), NTHRCNT 5411
Z BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NTHRCNT 5412
Z BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80), NTHRCNT 5413
Z QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NTHRCNT 5414
Z QTIPIN(400), QTIP(80), RHOVG(400), PEX(400), NTHRCNT 5415
Z BCTIME(50), TTIO(50), PTIO(50), WPLEN, NTHRCNT 5416
Z WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP NTHRCNT 5417
NTHRCNT 5418
COMMON /TCO/ ADUMP, BTA, CD, CP, NTHRCNT 5419
Z GAM, FIM, R, SPAN, TOG, NTHRCNT 5420
Z NDUMP, WIM, AKC(15,80), AKW(15,80), NTHRCNT 5421
Z A(400), AJET(80), AM2(80), CNUM(80), NTHRCNT 5422
Z DH(80), DHF(80), DHJ(80), NTHRCNT 5423
Z DLX(400), FF(80), HC(80), HG(80), NTHRCNT 5424
Z P(2,15,80), PEXIT(15), PUMP(80), OG(80), NTHRCNT 5425
Z QSNK(80), PR(80), S(15), T(2,15,400), NTHRCNT 5426
Z TG(80), TAU(400), WFC(80), NTHRCNT 5427
Z WJ(15,80), WCROS(2,15,80), XN(80), NTHRCNT 5428
Z ICOR, IFILM, IHUB, ITIP, NTHRCNT 5429
Z ISBLOK, ISLICE, NBLKSZ, NSLICE, NTHRCNT 5430
Z NFWD, NSTA, IHC(80) NTHRCNT 5431
NTHRCNT 5432
DO 1000 I = 1,NSTA NTHRCNT 5433
L = 5*I - 2 NTHRCNT 5434
LJ = L - 1 NTHRCNT 5435
LOUT = L - 2 NTHRCNT 5436
TC = (T(2,ISLICE,LJ) + T(2,ISLICE,LOUT))/2.0 - 460. NTHRCNT 5437
TW = T(2,ISLICE,L) - 460. NTHRCNT 5438
NTHRCNT 5439
LOOK UP COATING THERMAL CONDUCTIVITY IN TABLE AKCTBL. BTHRCNT 5440
NTHRCNT 5441
IF (TC.GT.AKCTBL(1)) GO TO 150 NTHRCNT 5442
NTHRCNT 5443
FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE NTHRCNT 5444
BELLOW TABLE NTHRCNT 5445
RATIO = (TC - AKCTBL(1))/(AKCTBL(3) - AKCTBL(1)) NTHRCNT 5446
AKC(ISLICE,I) = AKCTBL(2) + (AKCTBL(4)-AKCTBL(2))*RATIO NTHRCNT 5447
GO TO 500 NTHRCNT 5448
NTHRCNT 5449
150 CONTINUE NTHRCNT 5450
NTHRCNT 5451
C FIND SIZE OF TABLE NTHRCNT 5452
NTHRCNT 5453
DO 152 J = 3,19,2 NTHRCNT 5454
JLSI = J-1 NTHRCNT 5455
IF (AKCTBL(J).LE.0.1) GO TO 154 NTHRCNT 5456
152 CONTINUE NTHRCNT 5457
154 JLSI = JLSI-1 NTHRCNT 5458
NTHRCNT 5459
LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKCTBL. NTHRCNT 5460

```

```

C          NTHRCNT 5461
DO 170 J = 3,JLSTM,2      NTHRCNT 5462
IF (TC.GT.AKCTBL(J)) GO TO 160      NTHRCNT 5463
C          NTHRCNT 5464
C          NTHRCNT 5465
C          NTHRCNT 5466
C          NTHRCNT 5467
C          NTHRCNT 5468
C          NTHRCNT 5469
C          NTHRCNT 5470
C          NTHRCNT 5471
C          NTHRCNT 5472
C          NTHRCNT 5473
C          NTHRCNT 5474
C          NTHRCNT 5475
C          NTHRCNT 5476
C          NTHRCNT 5477
C          NTHRCNT 5478
C          NTHRCNT 5479
C          NTHRCNT 5480
C          NTHRCNT 5481
C          NTHRCNT 5482
C          NTHRCNT 5483
C          NTHRCNT 5484
C          NTHRCNT 5485
C          NTHRCNT 5486
C          NTHRCNT 5487
C          NTHRCNT 5488
C          NTHRCNT 5489
C          NTHRCNT 5490
C          NTHRCNT 5491
C          NTHRCNT 5492
C          NTHRCNT 5493
C          NTHRCNT 5494
C          NTHRCNT 5495
C          NTHRCNT 5496
C          NTHRCNT 5497
C          NTHRCNT 5498
C          NTHRCNT 5499
C          NTHRCNT 5500
C          NTHRCNT 5501
C          NTHPCNT 5502
C          NTHRCNT 5503
C          NTHRCNT 5504
C          NTHRCNT 5505
C          NTHPCNT 5506
C          NTHRCNT 5507
C          NTHRCNT 5508
C          NTHRCNT 5509
C          NTHFCNT 5510
C          NTHRCNT 5511
C          NTHRCNT 5512
C          NTHRCNT 5513
C          NTHRCNT 5514
C          NTHPCNT 5515
C          NTHRCNT 5516
C          NTHRCNT 5517
C          NTHRCNT 5518
C          NTHRCNT 5519
C          NTHRCNT 5520

C          FOUND LOCATION, NOW INTERPOLATE.

C          RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))
C          AKC(ISLICE,I) = AKCTBL(J-1) + (AKCTBL(J+1) - AKCTBL(J-1))*RATIO
C          GO TO 500
160      IF (J.LT.JLSTM) GO TO 170
C          TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.
C          RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))
C          AKC(ISLICE,I) = AKCTBL(J-1) + (AKCTBL(J+1) - AKCTBL(J-1))*RATIO
C          GO TO 500
170      CONTINUE
500      CONTINUE
C          NOW LOOK UP METAL CONDUCTIVITY IN TABLE AKWTBL.
C          IF (TW.GT.AKWTBL(1)) GO TO 550
C          FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE
C          BELOW TABLE
C          RATIO = (TW - AKWTBL(1))/(AKWTBL(3) - AKWTBL(1))
C          AKW(ISLICE,I) = AKWTBL(2) + (AKWTBL(4)-AKWTBL(2))*RATIO
C          GO TO 1000
C          550      CONTINUE
C          FIND SIZE OF TABLE
C          DO 552 J = 3,19,2
C          JLST = J-1
C          IF (AKWTBL(J).LE.0.1) GO TO 554
552      CONTINUE
554      JLSTM = JLST-1
C          LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKWTBL.
C          DO 570 J = 3,JLSTM,2
C          IF (TW.GT.AKWTBL(J)) GO TO 560
C          FOUND LOCATION, NOW INTERPOLATE.
C          RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))
C          AKW(ISLICE,I) = AKWTBL(J-1) + (AKWTBL(J+1) - AKWTBL(J-1))*RATIO
C          GO TO 1000
560      IF (J.LT.JLSTM) GO TO 570
C          TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.
C          RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))
C          AKW(ISLICE,I) = AKWTBL(J-1) + (AKWTBL(J+1) - AKWTBL(J-1))*RATIO
C          GO TO 1000
570      CONTINUE
1000     CONTINUE
      RETURN
      END

```

```

C----SOURCE.NWROTTT
      SUBROUTINE WROUT(IDELT,JS,DELTAN,IVERGE)
C
C- SOURCE.NWROTTT-----
C
C DUMR2 CARRIES THE IMPINGEMENT JET REYNOLDS NO. IN FROM HCCOOL.
C
C     COMMON /CHKHOL/ WCHK(80), WCHKDM
C
C     COMMON /FLMCOL/ RHOVGA(80), EG(80), XFC(80), FLMEFF(80),
C                      XMUC(80), FMES(80), REFC(80), NFCSUP(80)
C
C     COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),
C                      CPC(80), GAMC(80), DUMR1(80), DUMR2(80)
C
C     COMMON /FADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),
C                      PIN(15), TIN(15), WS
C
C     COMMON /TCO/ ADUMP, BTX, CD, CP,
C                      GAM, FIM, R, SPAN, TOG,
C                      WDUMP, WIM, AKC(15,80), AKW(15,80),
C                      A(400), AJET(80), AM2(80), CNUM(80),
C                      DH(80), DHF(80), DHJ(80),
C                      DLX(400), FP(80), HC(80),
C                      P(2,15,80), PEXIT(15), PUMP(80),
C                      QSNK(80), RR(80), S(15),
C                      TG(80), TAU(400), WFC(80),
C                      WJ(15,80), WCROS(2,15,80), XN(80),
C                      ICOR, IFILM, IHUB, ITIP,
C                      ISBLOK, ISLICE, NBLKSZ, NSLICE,
C                      NFWD, NSTA, IHC(80)
C
C     COMMON /TRNSNT/ RHOC, RHOM, SFHTC, SPHTM,
C                      DLTYME, TYME, TEPS, TYMMAX
C
C     COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),
C                      CTMPF(2), CTCOM(2), CDEN(2), CSPHT(2), CGASC(2),
C                      CVISC(2), CRHOVG(2), IUNITS
C
C DIMENSION DUM1(10),DUM2(10),DELTAN(15)
CONTINUE
10   IF (ISLICE.EQ.1) TBULK = 0.0
    IF (ISLICE.EQ.1) TOTSEN = 0.0
    TTYME = TYME
    IF (TTYME.LT.0.) TTYME=0.0
    WRITE(6,90) TTYME,DLTYME,WS
90   FORMAT(1H1,10X,' TIME = ',F6.2,' SEC., STEP SIZE = ',F6.3,
          ' SEC., WHEEL SPEED = ',F8.1,' RPM')
8805  WRITE(6,8806) ISLICE,IDEIT,JS,DELTAN(ISLICE),IVERGE
    ITRBG = NPWD + 2
    WRITE(6,100) ITRBG
C
    IP (IUNITS.EQ.2) WRITE(6,270)
    IP (IUNITS.EQ.1) WRITE(6,271)
    DO 210 I = 1,NSTA,2
    II = I
    LCOOL = 5*II
    NOS = LCOOL - 4
    DO 205 J = 1,4
    JM = NOS+J-1
      NWROTTT 5521
      NWROTTT 5522
      NWROTTT 5523
      NWROTTT 5524
      NWROTTT 5525
      NWROTTT 5526
      NWROTTT 5527
      NWROTTT 5528
      NWROTTT 5529
      NWROTTT 5530
      NWROTTT 5531
      NWROTTT 5532
      NWROTTT 5533
      NWROTTT 5534
      NWROTTT 5535
      NWROTTT 5536
      NWROTTT 5537
      NWROTTT 5538
      NWROTTT 5539
      NWROTTT 5540
      NWROTTT 5541
      NWROTTT 5542
      NWROTTT 5543
      NWROTTT 5544
      NWROTTT 5545
      NWROTTT 5546
      NWROTTT 5547
      NWROTTT 5548
      NWROTTT 5549
      NWROTTT 5550
      NWROTTT 5551
      NWROTTT 5552
      NWROTTT 5553
      NWROTTT 5554
      NWROTTT 5555
      NWROTTT 5556
      NWROTTT 5557
      NWROTTT 5558
      NWROTTT 5559
      NWROTTT 5560
      NWROTTT 5561
      NWROTTT 5562
      NWROTTT 5563
      NWROTTT 5564
      NWROTTT 5565
      NWROTTT 5566
      NWROTTT 5567
      NWROTTT 5568
      NWROTTT 5569
      NWROTTT 5570
      NWROTTT 5571
      NWROTTT 5572
      NWROTTT 5573
      NWROTTT 5574
      NWROTTT 5575
      NWROTTT 5576
      NWROTTT 5577
      NWROTTT 5578
      NWROTTT 5579
      NWROTTT 5580

```

```

DUM1(J) = T(2,ISLICE,JM) - 460.
IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)
205 CONTINUE
DUM1(5) = T(2,ISLICE,LCOOL) - 460.
IF (IUNITS.EQ.1) DUM1(5) = T(2,ISLICE,LCOOL)/CTMPF(1)
DUM1(6) = P(2,ISLICE,II)/CPRS(IUNITS)
DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)** (GAMC(II)/
Z (GAMC(II)-1.))
DUM1(8) = HC(II)/CHTC(IUNITS)
DUM1(9) = HG(II)/CHTC(IUNITS)
DUM1(10)= TG(II) - 460.
IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8
IF (BTA.GT..001) DUM1(10) = 9.E20
IF (BTA.GT..001) DUM1(9) = 9.E20
WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))
IF (I.EQ.NFWD) WRITE(6,276)
210 CONTINUE
IF (IUNITS.EQ.2) WRITE(6,278)
IF (IUNITS.EQ.1) WRITE(6,279)
DO 220 I = 1,NSTA,2
II = I
LCOOL = 5*II
NOS = LCOOL - 4
DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)
DUM2(2) = DUMR2(II)
DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)
DUM2(4) = RE(II)
DUM2(5) = SQRT(AM2(II))
DUM2(6) = FF(II)
DUM2(7) = WFC(II)/CMSFL(IUNITS)
DUM2(8) = FLMEFF(II)
WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8))
IF (I.EQ.NFWD) WRITE(6,276)
220 CONTINUE
DUM2(9) = WDUMP/CMSFL(IUNITS)
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM
C
ITRBG = NPWD + 1
WRITE(6,124) ISLICE,ITRBG
C
IF (IUNITS.EQ.2) WRITE(6,270)
IF (IUNITS.EQ.1) WRITE(6,271)
DO 230 I = 1,NSTA,2
II = I
IF (I.GT.1) II = I-1
LCOOL = 5*II
NOS = LCOOL - 4
DO 225 J = 1,5
JM = NOS+J-1
DUM1(J) = T(2,ISLICE,JM) - 460.
IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)
225 CONTINUE
DUM1(6) = P(2,ISLICE,II)/CPRS(IUNITS)
DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)** (GAMC(II)/
Z (GAMC(II)-1.))
DUM1(8) = HC(II)/CHTC(IUNITS)
DUM1(9) = HG(II)/CHTC(IUNITS)
DUM1(10)= TG(II) - 460.
IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8
NWROTTT 5581
NWROTTT 5582
NWROTTT 5583
NWROTTT 5584
NWROTTT 5585
NWROTTT 5586
NWROTTT 5587
NWROTTT 5588
NWROTTT 5589
NWROTTT 5590
NWROTTT 5591
NWROTTT 5592
NWROTTT 5593
NWROTTT 5594
NWROTTT 5595
NWROTTT 5596
NWROTTT 5597
NWROTTT 5598
NWROTTT 5599
NWROTTT 5600
NWROTTT 5601
NWROTTT 5602
NWROTTT 5603
NWROTTT 5604
NWROTTT 5605
NWROTTT 5606
NWROTTT 5607
NWROTTT 5608
NWROTTT 5609
NWROTTT 5610
NWROTTT 5611
NWROTTT 5612
NWROTTT 5613
NWROTTT 5614
NWROTTT 5615
NWROTTT 5616
NWROTTT 5617
NWROTTT 5618
NWROTTT 5619
NWROTTT 5620
NWROTTT 5621
NWROTTT 5622
NWROTTT 5623
NWROTTT 5624
NWROTTT 5625
NWROTTT 5626
NWROTTT 5627
NWROTTT 5628
NWROTTT 5629
NWROTTT 5630
NWROTTT 5631
NWROTTT 5632
NWROTTT 5633
NWROTTT 5634
NWROTTT 5635
NWROTTT 5636
NWROTTT 5637
NWROTTT 5638
NWROTTT 5639
NWROTTT 5640

```

```

IF (BTA.GT..001) DUM1(10) = 9.E20          NWROTTT 5641
IF (BTA.GT..001) DUM1(9) = 9.E20          NWROTTT 5642
WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))  NWROTTT 5643
IF (I.EQ.NFWD) WRITE(6,276)                NWROTTT 5644
230  CONTINUE                                NWROTTT 5645
IF (IUNITS.EQ.2) WRITE(6,278)                NWROTTT 5646
IF (IUNITS.EQ.1) WRITE(6,279)                NWROTTT 5647
DO 240 I = 1,NSTA,2                         NWROTTT 5648
II = I                                       NWROTTT 5649
IF (I.GT.1) II = I-1                         NWROTTT 5650
LCOOL = 5*II                                 NWROTTT 5651
NOS = LCOOL - 4                            NWROTTT 5652
DUM2(1) = WJ(ISLICE,II)/CMSPL(IUNITS)       NWROTTT 5653
DUM2(2) = DUMR2(II)                         NWROTTT 5654
DUM2(3) = WCROS(2,ISLICE,II)/CMSPL(IUNITS) NWROTTT 5655
DUM2(4) = RE(II)                            NWROTTT 5656
DUM2(5) = SQRT(AM2(II))                     NWROTTT 5657
DUM2(6) = FF(II)                           NWROTTT 5658
DUM2(7) = WPC(II)/CMSPL(IUNITS)             NWROTTT 5659
DUM2(8) = FLMEFF(II)                        NWROTTT 5660
WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8)) NWROTTT 5661
IF (I.EQ.NFWD) WRITE(6,276)                  NWROTTT 5662
240  CONTINUE                                NWROTTT 5663
DUM2(9) = WDUMP/CMSPL(IUNITS)              NWROTTT 5664
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM NWROTTT 5665
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM NWROTTT 5666
C
1000  CONTINUE                                NWROTTT 5667
C
C   TO DETERMINE THE MEAN OUTSIDE SURFACE TEMPERATURE FOR EACH
C   SIDE OF THE BLADE, AND                               NWROTTT 5668
C   TO LOCATE THE EXTREME TEMPERATURE POINTS, BOTH HIGH AND LOW. NWROTTT 5669
C
XTOT = 0.                                     NWROTTT 5670
XTOTMD = 0.0                                   NWROTTT 5671
HBAR = HC(1)*(DLX(2)+DLX(3))/2.               NWROTTT 5672
TBAR = T(2,ISLICE,1)*(DLX(6)+DLX(11))/2.     NWROTTT 5673
TBARMD = T(2,ISLICE,3)*(DLX(8)+DLX(13))/2.   NWROTTT 5674
ISTAT = 1                                      NWROTTT 5675
DO 1004 I = 2,NSTA                           NWROTTT 5676
NODM = 5*I-2                                  NWROTTT 5677
ISTAT = ISTAT + 5                            NWROTTT 5678
ISTATD = ISTAT + 10                          NWROTTT 5679
TBARMD = TBARMD + T(2,ISLICE,NODM)*(DLX(NODM)+DLX(NODM+10))/2. NWROTTT 5680
XTOTMD = XTOTMD + DLX(NODM)                 NWROTTT 5681
TBAR = TBAR + T(2,ISLICE,ISTAT)*(DLX(ISTAT)+DLX(ISTATD))/2. NWROTTT 5682
XTOT = XTOT + DLX(ISTAT)                   NWROTTT 5683
HBAR = HBAR + HC(1)*(DLX(ISTAT)+DLX(ISTATD))/2. NWROTTT 5684
1004  CONTINUE                                NWROTTT 5685
IF (IUNITS.EQ.1) TBAR = TBAR/(1.8*XTOT)      NWROTTT 5686
IF (IUNITS.EQ.2) TBAR = TBAR/XTOT - 460.      NWROTTT 5687
IF (IUNITS.EQ.1) TBARMD = TBARMD/(1.8*XTOT)  NWROTTT 5688
IF (IUNITS.EQ.2) TBARMD = TBARMD/XTOTMD - 460. NWROTTT 5689
TBULK = TBULK + TBARMD*S(ISLICE)            NWROTTT 5690
TOTSPN = TOTSPN + S(ISLICE)                  NWROTTT 5691
HBAR = HBAR/(XTOT*CHTC(IUNITS))             NWROTTT 5692
1008  IF (IUNITS.EQ.2) WRITE(6,1115) TBAR,TBARMD,HBAR NWROTTT 5693
      IF (IUNITS.EQ.1) WRITE(6,1116) TBAR,TBARMD,HBAR NWROTTT 5694
C
      TSMAX = T(2,ISLICE,1) - 460.              NWROTTT 5695

```

```

TSMIN = T(2,ISLICE,1) - 460.
TPMAX = T(2,ISLICE,1) - 460.
TPMIN = T(2,ISLICE,1) - 460.
ISUCMX = 1
ISUCMN = 1
IPRSMX = 1
IPRSMN = 1
IPRES = 1
ISUCT = -4

C
DO 1080 I = 3,NSTA,2
IPRES = IPRES + 10
ISUCT = ISUCT + 10

C
C
IF (T(2,ISLICE,IPRES)-460..GT.TPMAX) GO TO 1030
IF (T(2,ISLICE,IPRES)-460..LT.TPMIN) GO TO 1040
GO TO 1050

C
1030 TPMAX = T(2,ISLICE,IPRES)-460.
IPRSMX = I
GO TO 1050

C
1040 TPMIN = T(2,ISLICE,IPRES)-460.
IPRSMN = I

C
1050 IF (T(2,ISLICE,ISUCT)-460..GT.TSMAX) GO TO 1060
IF (T(2,ISLICE,ISUCT)-460..LT.TSMIN) GO TO 1070
GO TO 1080

C
1060 TSMAX = T(2,ISLICE,ISUCT)-460.
ISUCMX = I - 1
GO TO 1080

C
1070 TSMIN = T(2,ISLICE,ISUCT)-460.
ISUCMN = I - 1
1080 CONTINUE

C
C
IF (ISLICE.LT.NSLICE) GO TO 1095
TBULK = TBULK/TOTSPN
IF (IUNITS.EQ.1) WRITE(6,1091) TYME,TBULK
IF (IUNITS.EQ.2) WRITE(6,1090) TYME,TBULK

C
1090 FORMAT(1H2,30X,'TIME =',F6.3,' SEC., OVERALL BULK TEMPERATURE =',
Z F7.1,' F')
1091 FORMAT(1H2,30X,'TIME =',F6.3,' SEC., OVERALL BULK TEMPERATURE =',
Z F7.1,' K')
1095 CONTINUE
1115 FORMAT(/' OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T =',F7.1,
Z ' F, MID-WALL T =',F7.1,
Z ' F, COOLANT H = ',F6.1,' BTU/(HR/FT**2/R)')
1116 FORMAT(/' OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T =',F7.1,
Z ' K, MID-WALL T =',F7.1,
Z ' K, COOLANT H = ',F6.1,' WATTS/M**2/K')
IF (IUNITS.EQ.2) WRITE(6,1120) TPMAX,IPRSMX,TPMIN,IPRSMN,TSMAX,
Z ISUCMX,TSMIN,ISUCMN
1120 FORMAT( /12X,'EXTREMES OF OUTER SURFACE TEMPERATURES (F) '/6X,
Z 'PRESSURE SIDE: ',
Z F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X,
Z F7.1,' ')

```

```

Z      'SUCTION SIDE: ',                                NWROTTT 5761
Z      F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2)  NWROTTT 5762
IF (IUNITS.EQ.2) GO TO 1130                           NWROTTT 5763
TPMAX = (TPMAX+460.)/1.8                               NWROTTT 5764
TPMIN = (TPMIN+460.)/1.8                               NWROTTT 5765
TSMAX = (TSMAX+460.)/1.8                               NWROTTT 5766
TSMIN = (TSMIN+460.)/1.8                               NWROTTT 5767
WRITE(6,1125) TPMAX,IPRSMX,TPMIN,IPRSMN,TSMAX,ISUCMX,TSMIN,ISUCMN NWROTTT 5768
1125 FORMAT( /12X,'EXTREMES OF OUTER SURFACE TEMPERATURES (K)'/6X, NWROTTT 5769
Z      'PRESSURE SIDE: ',                                NWROTTT 5770
Z      F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X, NWROTTT 5771
Z      'SUCTION SIDE: ',                                NWROTTT 5772
Z      F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2)  NWROTTT 5773
1130 CONTINUE                                           NWROTTT 5774
100 FORMAT(30X,'PRESSURE SIDE , TRAILING EDGE REGION BEGINS AT ', NWROTTT 5775
Z      'STATION-',I3)                                 NWROTTT 5776
124 FORMAT(1H1,/' SLICE NO. ',I2,17X,'SUCTION SIDE , TRAILING EDGE ', NWROTTT 5777
Z      'REGION BEGINS AT STATION-',I3)                NWROTTT 5778
8806 FORMAT(' SLICE NO.',I2,', FLOW SPLIT NO.',I3,', SPLIT AT ', NWROTTT 5779
Z      'STATION',I3,                                     NWROTTT 5780
Z      ';' FRACTION SPLIT TO SUCTION SIDE IS',F7.4,I6,' ITERATIONS') NWROTTT 5781
270 FORMAT(/                                              NWROTTT 5782
Z      ' STATION*COOLANT* OUTSIDE *INTERFACE* MID-WALL* INSIDE * ', NWROTTT 5783
Z      'COOLANT * STATIC P* TOTAL P * HC,BTU/HR* HG,BTU/HR* TG' NWROTTT 5784
Z//' NUMBER *NODE NO* T (F) * T (F) * T (F) * T (F) * T ', NWROTTT 5785
Z      '(F) * (PSIA) * (PSIA) * /FT**2/R * /FT**2/R * (F)' NWROTTT 5786
Z/119('*')/)                                         NWROTTT 5787
271 FORMAT(/                                              NWROTTT 5788
Z      ' STATION*COOLANT* OUTSIDE *INTERFACE* MID-WALL* INSIDE * ', NWROTTT 5789
Z      'COOLANT * STATIC P* TOTAL P * HC * HG * TG' NWROTTT 5790
Z//' NUMBER *NODE NO* T (K) * T (K) * T (K) * T (K) * T ', NWROTTT 5791
Z      '(K) * (KPA) * (KPA) * W/M**2/K * W/M**2/K * (K)' NWROTTT 5792
Z/119('*')/)                                         NWROTTT 5793
274 FORMAT(I6,2X,I6,1X,7F10.1,3F11.1)                 NWROTTT 5794
276 FORMAT(47X,'BEGIN TRAILING EDGE REGION')          NWROTTT 5795
278 FORMAT(1H2,/' STATION * COOLANT * IMP. FLOW * RE-NO. * CROSSFLOW',NWROTTT 5796
Z      ' * RE-NO. * MACH NO., *'                         NWROTTT 5797
Z      ' FRICITION * FILM FLOW * EFFECTIVENESS *'/ NWROTTT 5798
Z      ' NUMBER * NODE NO * (LBM/SEC) * JET * (LBM/SEC) ',NWROTTT 5799
Z      '** CROSSFLOW * CROSSFLOW *'                      NWROTTT 5800
Z      ' FACTOR * (LBM/SEC) */115('*')/19X,'**',20X,'**',46X,'**') NWROTTT 5801
279 FORMAT(1H2,/' STATION * COOLANT * IMP. FLOW * RE-NO. * CROSSFLOW',NWROTTT 5802
Z      ' * RE-NO. * MACH NO., *'                         NWROTTT 5803
Z      ' FRICITION * FILM FLOW * EFFECTIVENESS *'/ NWROTTT 5804
Z      ' NUMBER * NODE NO * (KG/SEC) * JET * (KG/SEC) ',NWROTTT 5805
Z      '** CROSSFLOW * CROSSFLOW *'                      NWROTTT 5806
Z      ' FACTOR * (KG/SEC) */115('*')/19X,'**',20X,'**',46X,'**') NWROTTT 5807
280 FORMAT(1H ,I5,5X,I5,' **',F9.6,A1,F9.1,' * ',F9.6,2X,F9.1,4X,F9.6,NWROTTT 5808
Z      ,2X,F9.6,' * ',F9.6,4X,F9.6)                   NWROTTT 5809
290 FORMAT(//15X,' FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION IS 'NWROTTT 5810
Z      ,F10.6,A1,' LBM/SEC')                           NWROTTT 5811
291 FORMAT(//15X,' FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION IS 'NWROTTT 5812
Z      ,F10.6,A1,' KG/SEC')                           NWROTTT 5813
      RETURN                                           NWROTTT 5814
      END                                             NWROTTT 5815

```

REFERENCES

1. Gaugler, Raymond E.: TACT1, A Computer Program for the Transient Thermal Analysis of a Cooled Turbine Blade or Vane Equipped with a Coolant Insert. I - Users Manual. NASA TP-1271, 1978.
2. Poferl, David J.; and Svehla, Roger A.: Thermodynamic and Transport Properties of Air and Its Products of Combustion with ASTM-A-1 Fuel and Natural Gas at 20, 30, and 40 Atmospheres. NASA TN D-7488, 1973.
3. Stollery, J. L.; and El-Ehwany, A. A. M.: A Note on the Use of a Boundary-Layer Model for Correlating Film-Cooling Data. Int. J. Heat Mass Transfer, vol. 8, 1965, pp. 55-65.

TABLE I. - SUBROUTINE CALLS AND COMMON BLOCKS

Subroutine name	Source module	COMMON blocks	Called subroutines	Calling subroutines	Subroutine name	Source module	COMMON blocks	Called subroutines	Calling subroutines
BLOCK DATA	NGASDAT	/GAAS/	NONE	NONE	PARRAY	NPARAYT	/MATRX/ /PRPS/ /TCO/ /TRNSNT/	NONE	TCOEF
FLOWS	NFLOEST	/CHKHOL/ /FLMCOL/ /FRIC/ /PRPS/ /TCO/ /TRNSNT/	GASTBL	TCOEF	PLNUM	NPLENMP	/RADL/ /TCO/ /TRNSNT/ /UNITS/	GASTBL	MAIN PROGRAM
FLSPLT	NFLSPLP	NONE	NONE	TCOEF	PLOTMF	NPLOTIT	/PRPS/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	NONE	TCOEF MAIN PROGRAM
GASTBL	NGASTB	/GAAS/	NONE	FLOWS HCFRCD HCOOL HCPINS PLNUM TARRAY	PREP	NPREPAT	/BOUND/ /FLMCOL/ /FRIC/ /PRPS/ /SPECL/ /TCO/ /TRNSNT	NONE	INPRT MAIN PROGRAM
GAUSS	NGAUS	/MATRX/	NONE	TCOEF	TARRAY	NTARAYT	/BOUND/ /FLMCOL/ /MATRX/ /PRPS/ /TCO/ /TRNSNT/	GASTBL HCFRCD HCPINS	TCOEF
GETIN	NGETINT	/BOUND/ /FLMCOL/ /IMPCOR/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS	INPRT	MAIN PROGRAM	TCOEF	NTCOFTT	/MATRX/ /PRPS/ /TCO/ /TRNSNT/	FLows FLSPLT GAUSS HCOOL PARRAY PLOTMF TARRAY THRCON WROUT	MAIN PROGRAM
HCFRCD	NHCFRCT	/TCO/	GASTBL	HCOOL TARRAY	THRCON	NTHRCNT	/BOUND/ /TCO/	NONE	TCOEF
HCOOL	NHCOOLT	/IMPCOR/ /PRPS/ /TCO/	GASTBL HCFRCD	TCOEF	WROUT	NWROTT	/CHKHOL/ /FLMCOL/ /PRPS/ /RADL/ /TCO/ /TRNSNT/ /UNITS/	NONE	TCOEF
HCPINS	NHCPINT	/PRPS/ /TCO/	GASTBL	TARRAY					
INPRT	NINPRTT	/BOUND/ /GAAS/ /PRPS/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	PREP	GETIN					
MAIN PROGRAM	NTTACT	/BOUND/ /FLMCOL/ /GAAS/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	GETIN PLNUM PLOTMF PREP TCOEF	NONE					

TABLE II. - COMMON-BLOCK CROSS-REFERENCE TABLE

Subroutine	COMMON block												
	BOUND	CHKHOL	FLMCOL	FRIC	GAAS	IMPCOR	MATRX	PRPS	RADL	SPECL	TCO	TRNSNT	UNITS
BLOCK DATA					x								
FLOWS		x	x	x				x			x	x	
FLSPLT													
GASTBL					x								
GAUSS							x						
GETIN	x		x			x			x	x	x	x	x
HCFRCD												x	
HCOOL						x		x				x	
HCPINS								x				x	
INPRT	x				x			x	x	x	x	x	x
NTTACT	x		x		x				x	x	x	x	x
PARRAY							x	x			x	x	
PLNUM									x		x	x	x
PLOTMF								x		x	x	x	x
PREP	x		x	x				x		x	x	x	
TARRAY	x		x				x	x			x	x	
TCOEF							x	x			x	x	
THRCON	x										x		
WROUT		x	x					x	x		x	x	x

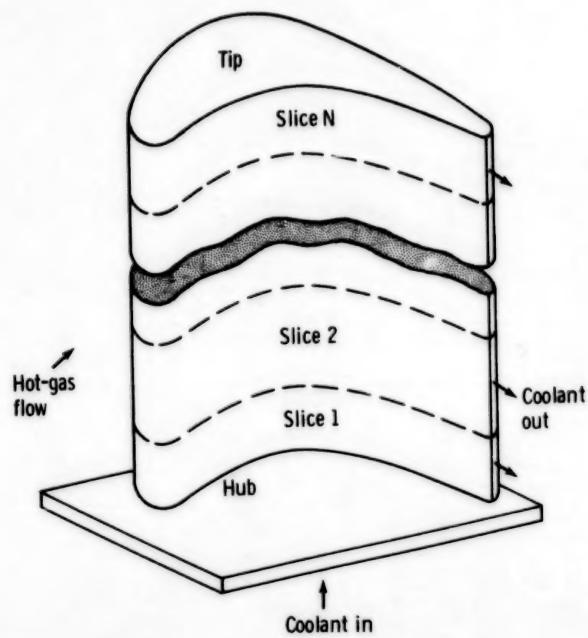


Figure 1. - Overall division of blade into slices.

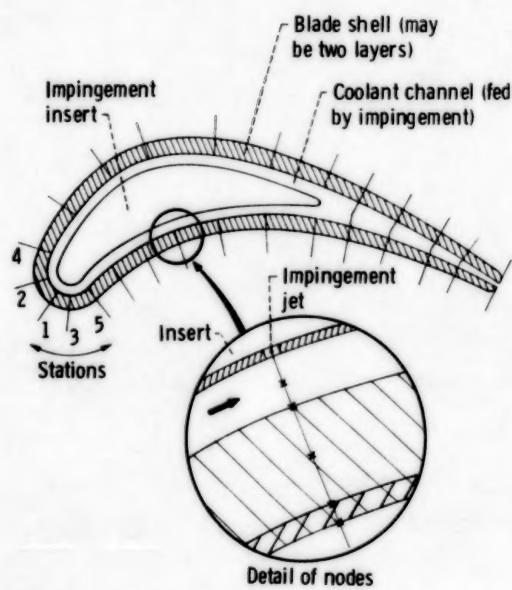


Figure 2 - Blade geometric model.

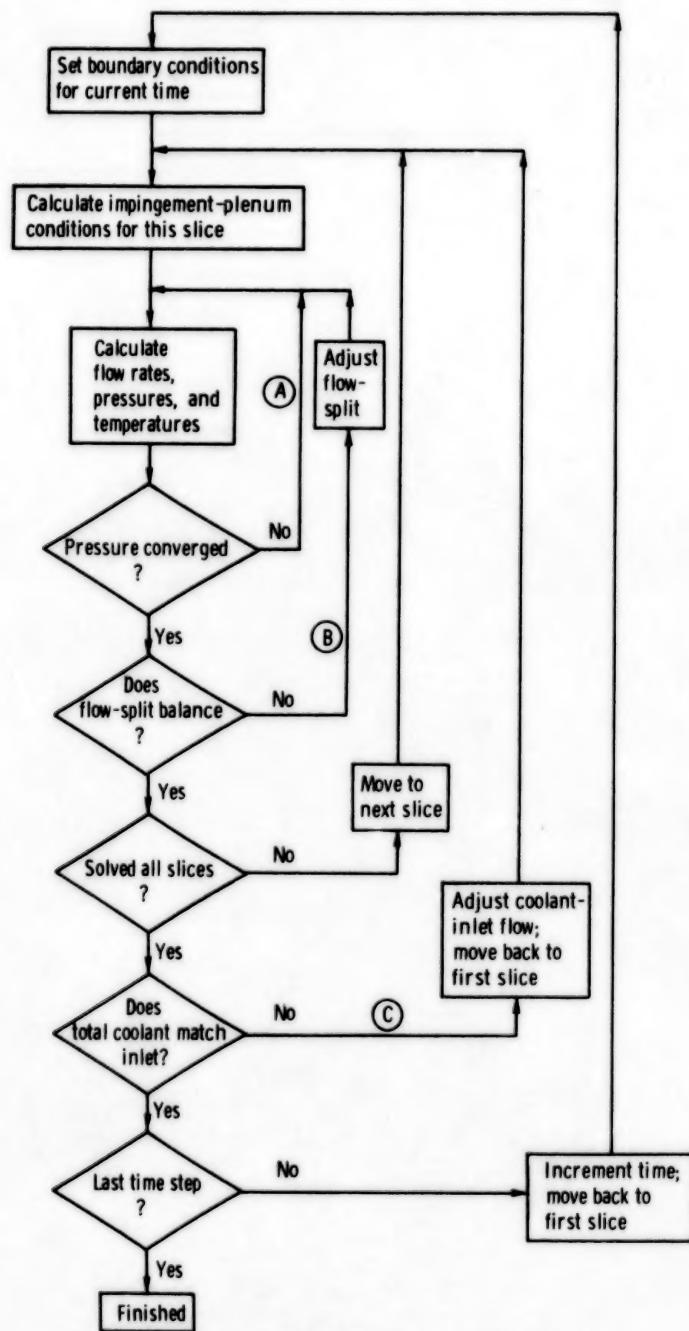


Figure 3. - Overall program procedure.

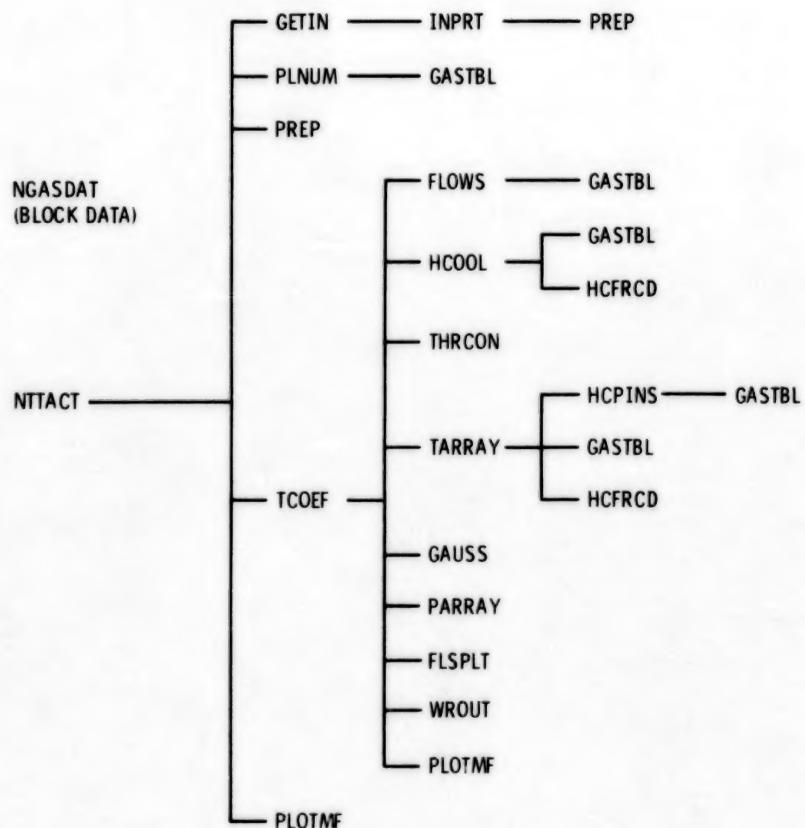


Figure 4. - Subroutine calling relations.

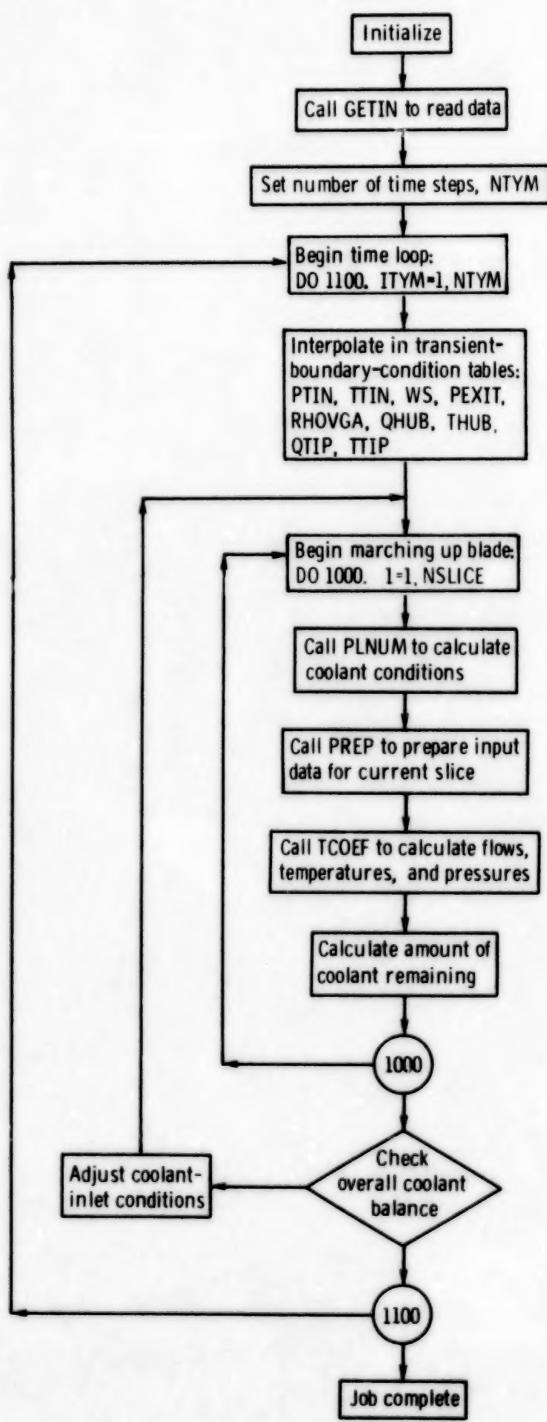


Figure 5. - Flow chart of main program.

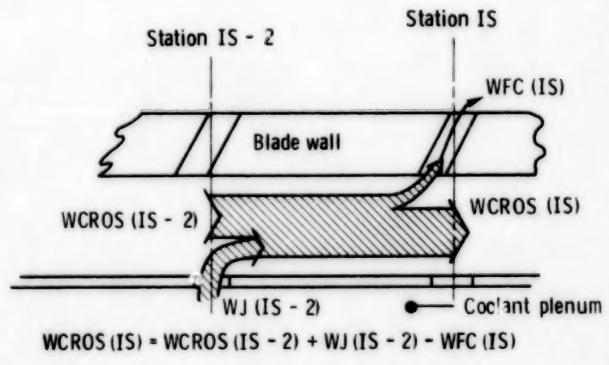


Figure 6. - Coolant-channel mass balance.

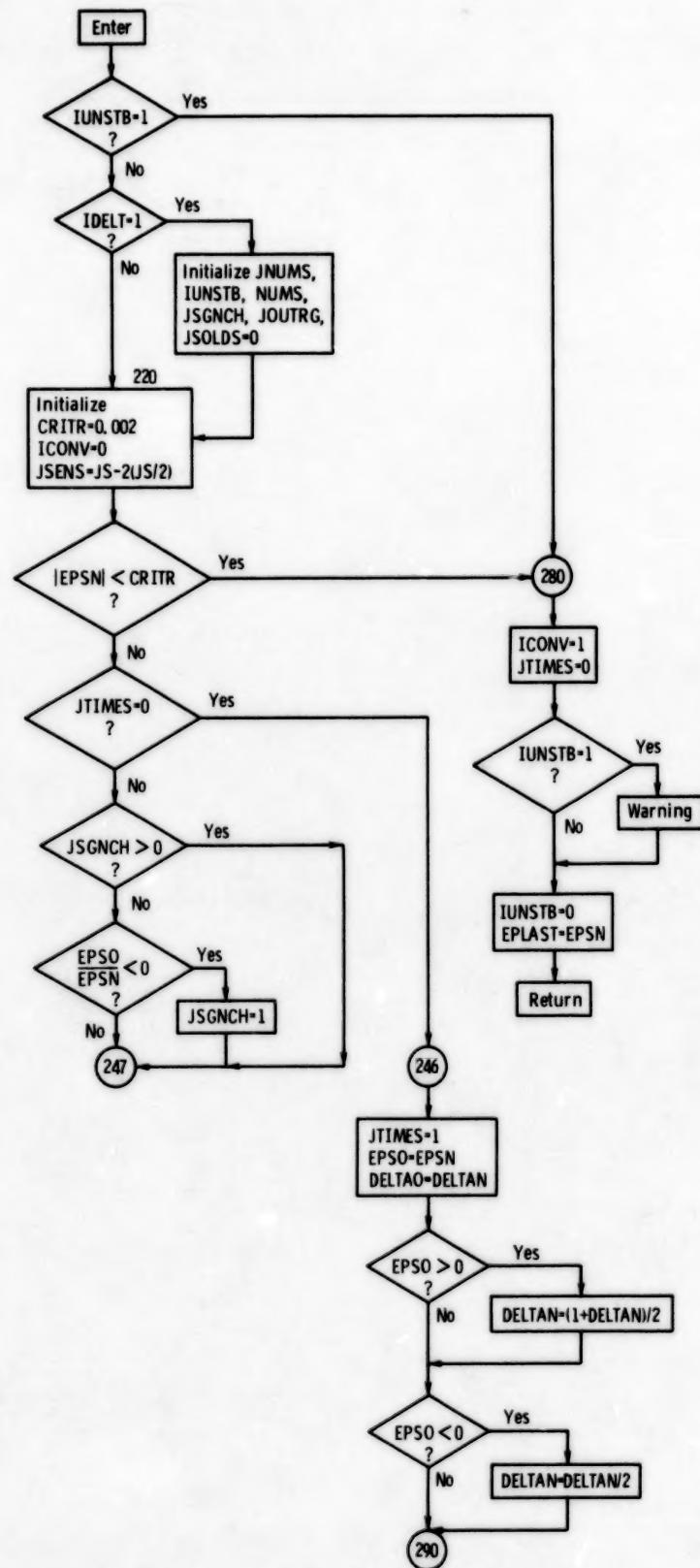


Figure 7. - Flow chart for subroutine FLSPLT.

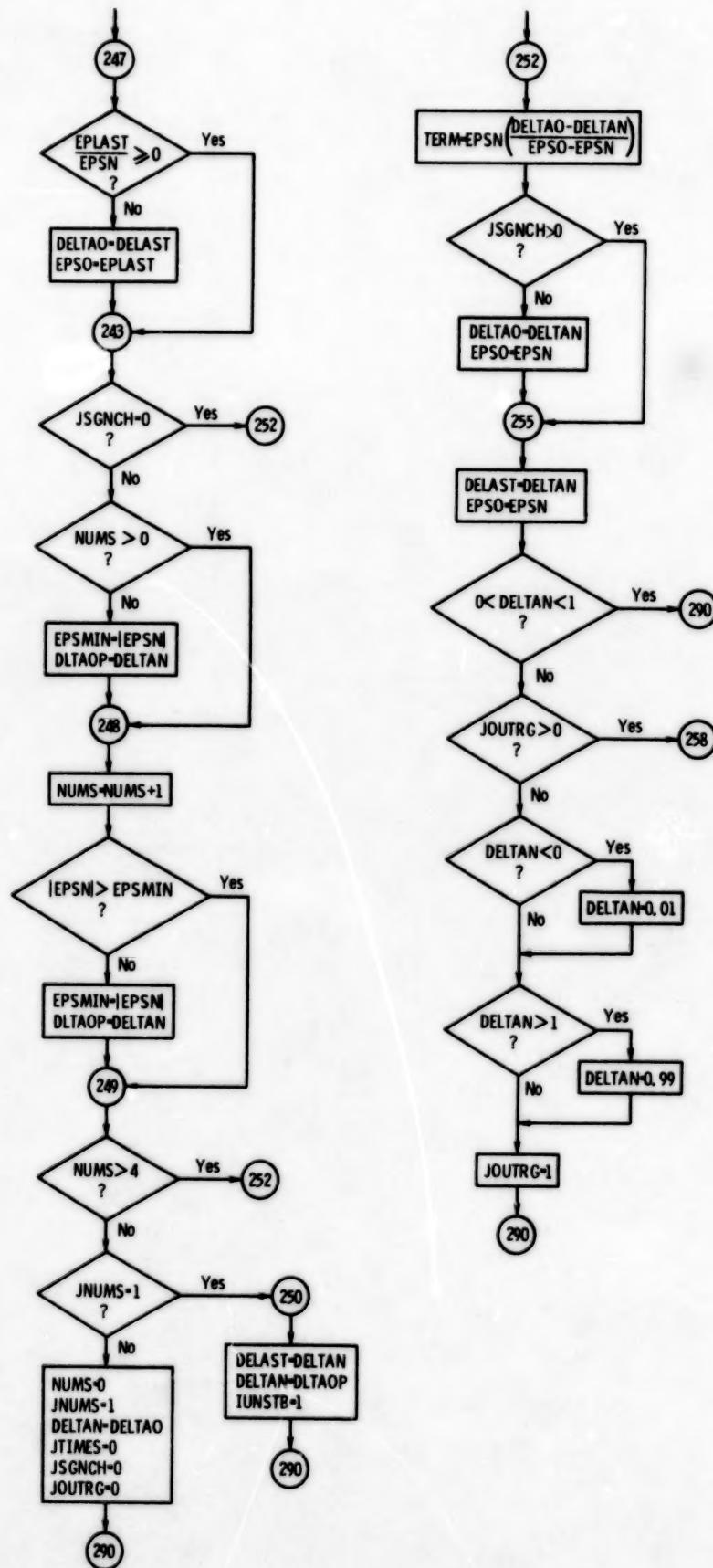


Figure 7. - Continued.

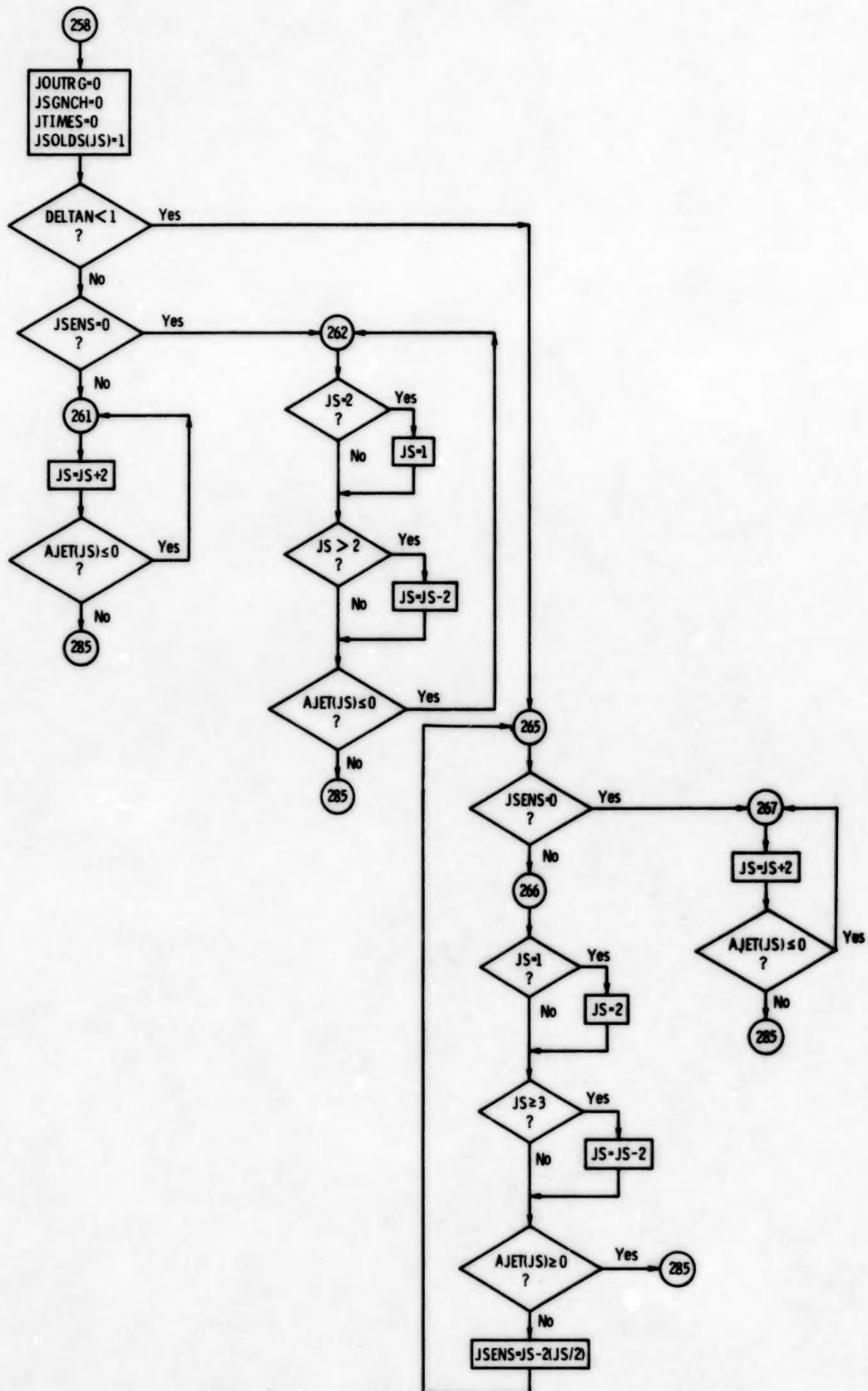


Figure 7. - Continued.

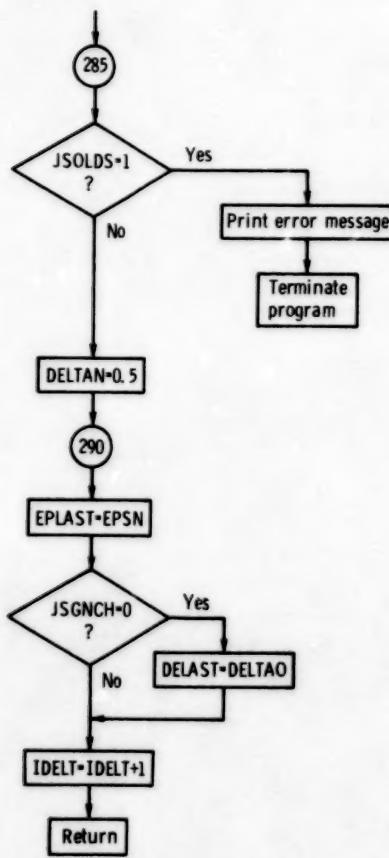


Figure 7. - Concluded.

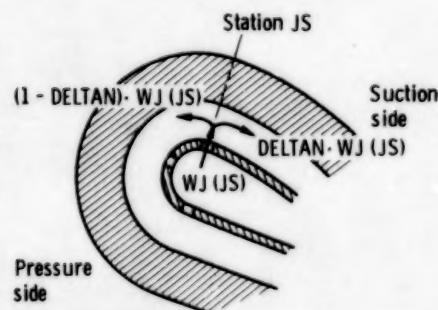
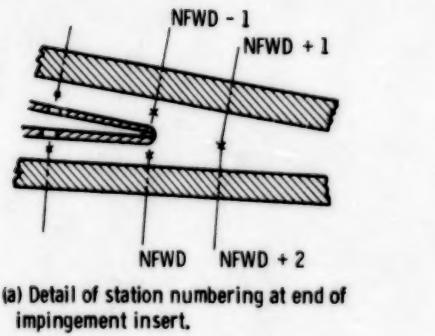
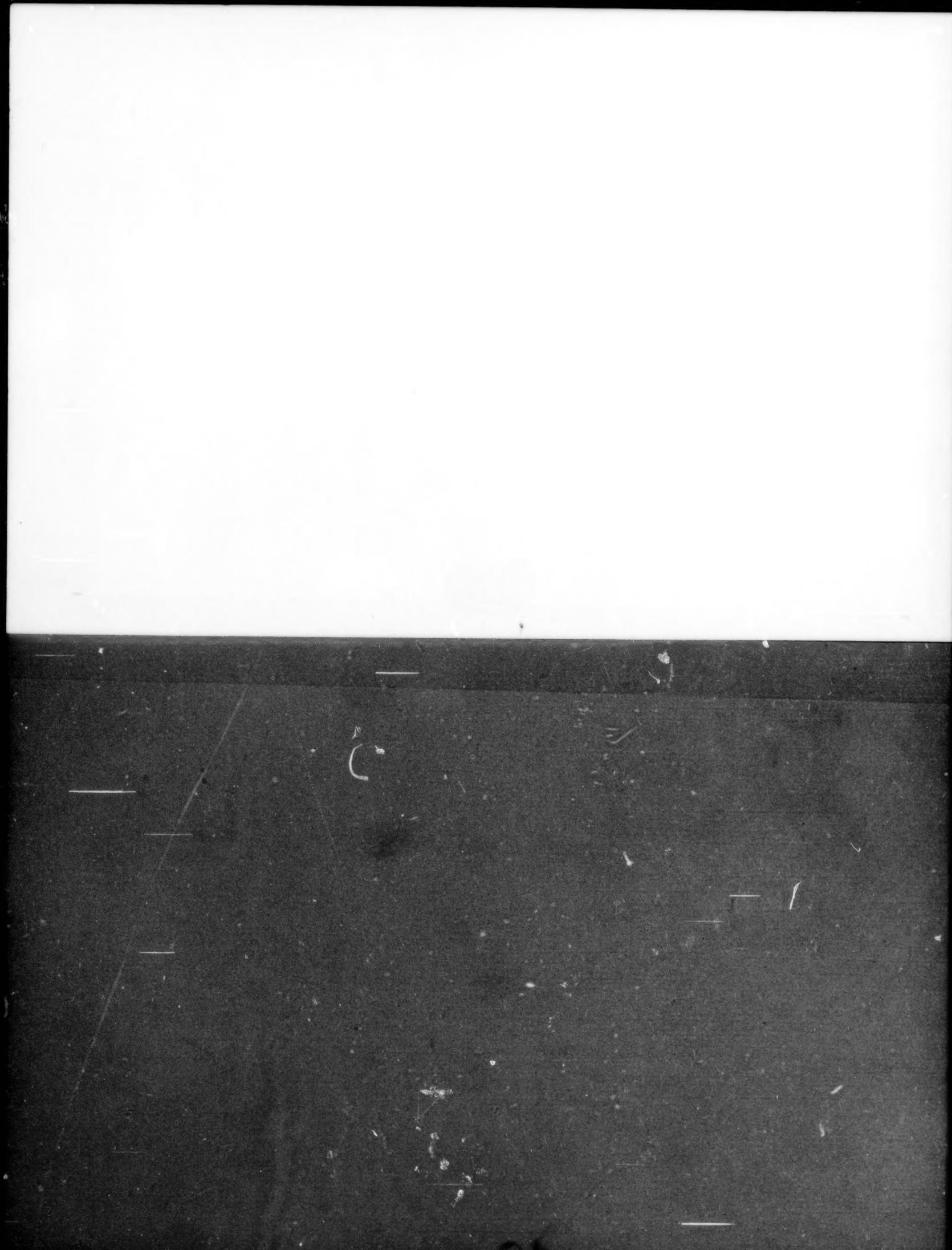


Figure 8. - Details of flow-split parameters.

1. Report No. NASA TP-1391	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle TACT1, A COMPUTER PROGRAM FOR THE TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE OR VANE EQUIPPED WITH A COOLANT INSERT. II - PROGRAMMERS MANUAL		5. Report Date January 1979	
7. Author(s) Raymond E. Gaugler		6. Performing Organization Code	
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135		8. Performing Organization Report No. E-9767	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		10. Work Unit No. 505-04	
15. Supplementary Notes		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Paper	
		14. Sponsoring Agency Code	
16. Abstract <p>A computer program to calculate transient and steady-state temperatures, pressures, and coolant flows in a cooled axial-flow turbine blade or vane with an impingement insert is described. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying either impingement or convection heat transfer at each internal flow station. Spent impingement air flows in a chordwise direction and is discharged through the trailing edge and through film-cooling holes. The ability of the program to handle film cooling is limited by the internal flow model. Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The blade wall can have two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.</p>			
17. Key Words (Suggested by Author(s)) Heat transfer Turbine cooling Computer program Impingement cooling		18. Distribution Statement Unclassified - unlimited STAR Category 34	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 163	22. Price* A08

* For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1979



END

September 17, 1979